

Amazon EC2 Container Service

(AKA EC2 ECS)

(AKA ECS)

Brian Nuskowski

@nuskowski

Github: broamski



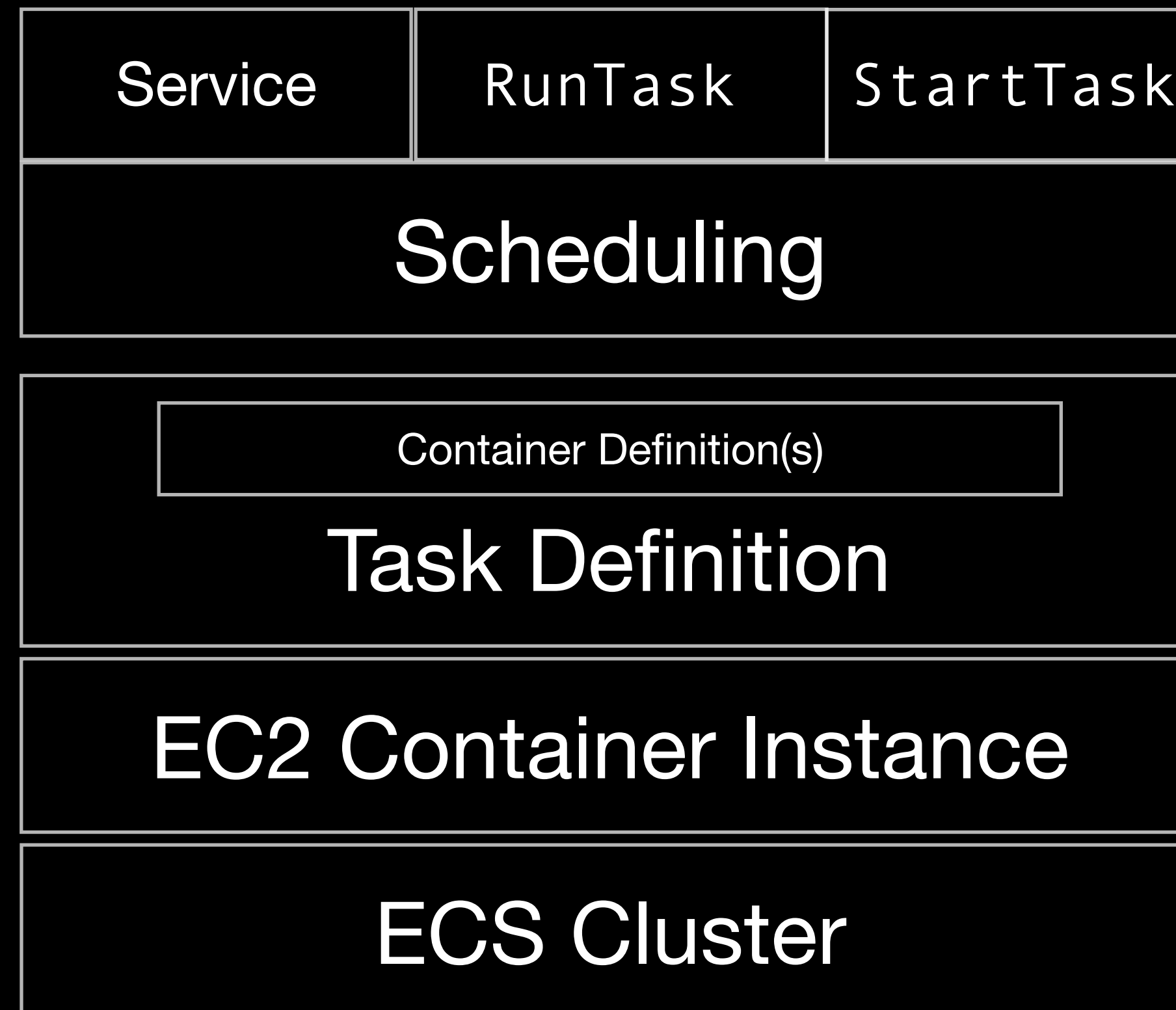
Amazon EC2 Container Registry

(AKA EC2 ECR)

(AKA ECR)

The Problem Statements

- How do I orchestrate and schedule them?
- Where do I store them?



ECS Cluster

 ECS Cluster

EC2 Container Instance

ECS Cluster



EC2 Container Instances



EC2 Instance

EC2 Container Instance Criteria

1. Linux
2. Docker Daemon
3. ECS Agent

ECS Cluster - Building a Container Instance Fleet

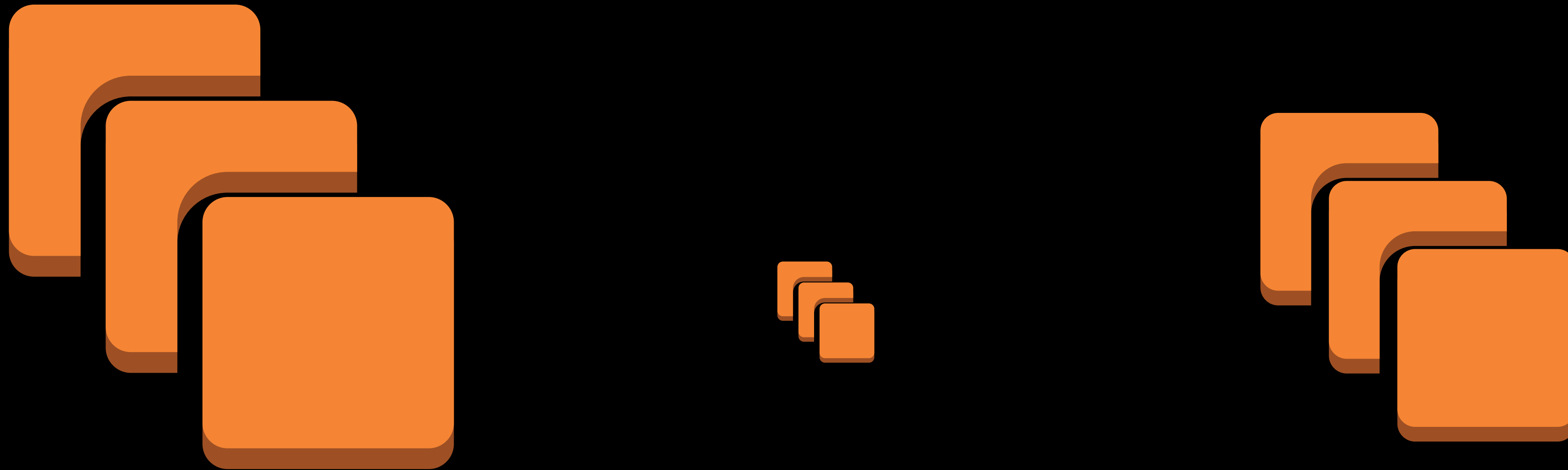
Amazon ECS-optimized AMI (`amzn-ami-2016.03.i-amazon-ecs-optimized`) consists:

- The latest minimal version of the Amazon Linux AMI
- The latest version of the Amazon ECS container agent (1.12.2)
- The recommended version of Docker for the latest Amazon ECS container agent (1.11.2)
- The latest version of the `ecs-init` package to run and monitor the Amazon ECS agent (1.12.2-1)

ECS Cluster - Building a Container Instance Fleet

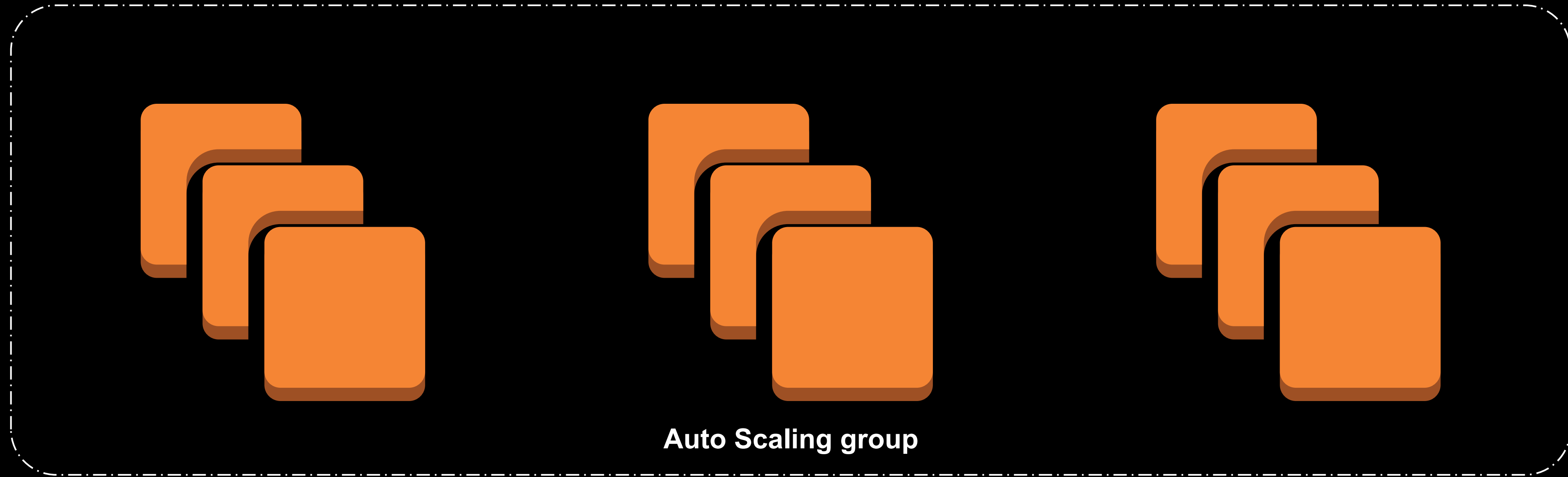
```
#!/bin/bash
# install docker
# do your other things
docker run --name ecs-agent \
  --detach=true \
  --restart=on-failure:10 \
  --volume=/var/run/docker.sock:/var/run/docker.sock \
  --volume=/var/log/ecs:/log \
  --volume=/var/lib/ecs/data:/data \
  --net=host \
  --env=ECS_LOGFILE=/log/ecs-agent.log \
  --env=ECS_LOGLEVEL=info \
  --env=ECS_DATADIR=/data \
  --env=ECS_CLUSTER=<cluster_name> \
  --env=ECS_ENABLE_TASK_IAM_ROLE=true \
  --env=ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST=true \
  amazon/amazon-ecs-agent:<version or latest>
```

ECS Cluster - Building a (Bespoke) Container Instance Fleet



**Mix and
Match**

ECS Cluster - Building a (Robust) Container Instance Fleet



ECS Cluster - **Task Definition**

nginx:1.10.1

Container Definition

yourpythonapp:2.6.9

Container Definition

Task Definition: **WebApp**

- 1 or many Container Definitions per Task Definition
- Each container definition defines typical container parameters: volumes, ports exposed, metadata, etc.
- Task Definitions are immutable

ECS Cluster - **Task Definition**

nginx:1.10.1

Container Definition

redis:latest

Container Definition

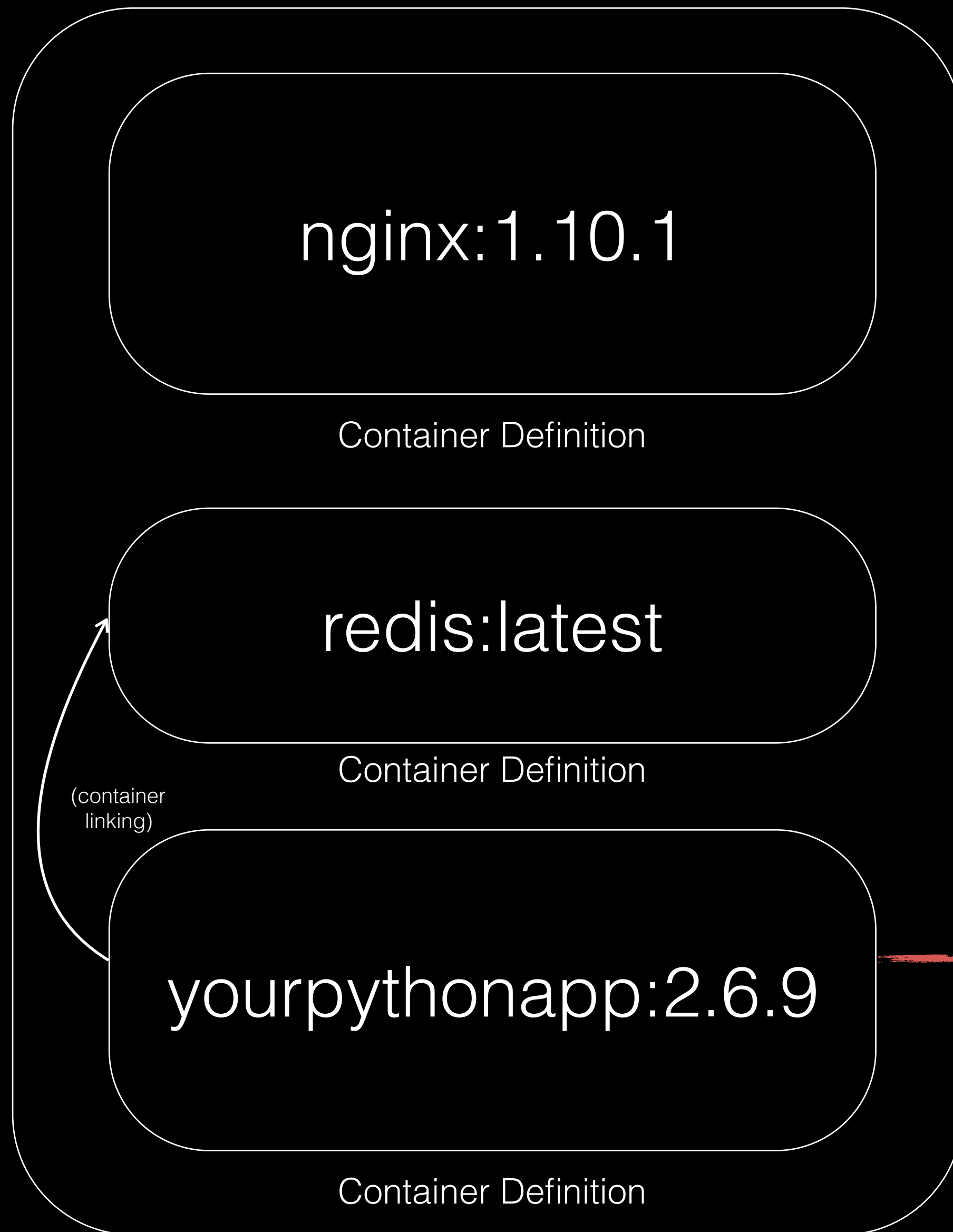
yourpythonapp:2.6.9

Container Definition

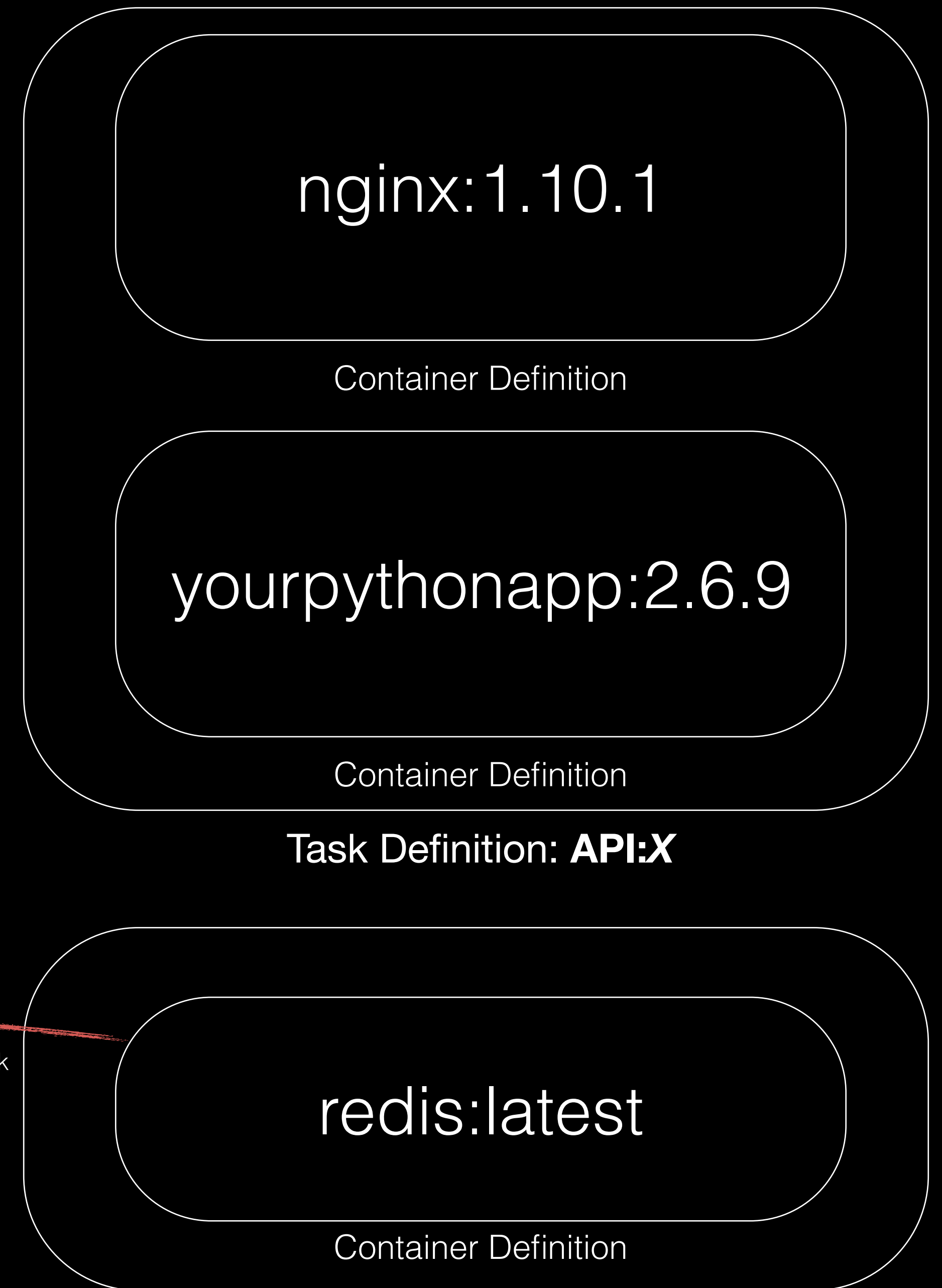
Task Definition: **WebApp**

- 1 or many Container Definitions per Task Definition
- Each container definition defines typical container parameters: volumes, ports exposed, metadata, etc.
- Task Definitions are immutable

Task Definitions



Task Definition: **WebApp:X** (Family:Revision)



Task Definition: **Cache:X**



Task Definitions

nginx:1.10.1

Container Definition

Task Definition: **nginx:1**

redis:latest

Container Definition

Task Definition: **redis:1**

pythonapp:2.4.3

Container Definition

Task Definition: **app:1**

Task Definitions

nginx:1.10.1

Container Definition

redis:latest

Container Definition

yourpythonapp:2.6.9

Container Definition

Task Definition: **WebApp:1**



nginx:1.10.1

Container Definition

redis:latest

Container Definition

yourpythonapp:2.7.0

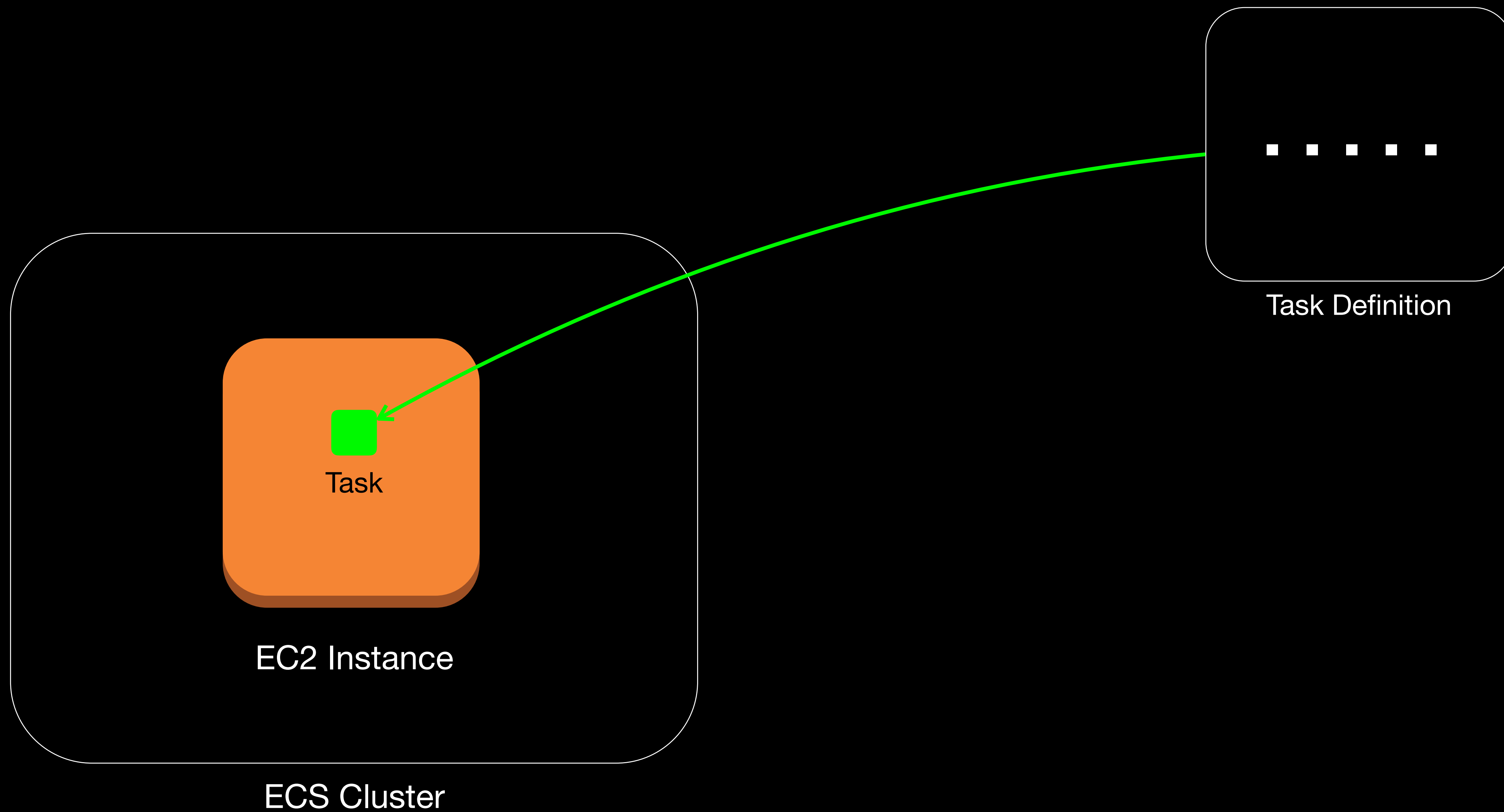
Container Definition

Task Definition: **WebApp:2**

(container linking)

(container linking)

Scheduling Tasks



Scheduling Tasks

Continuous
(Service)

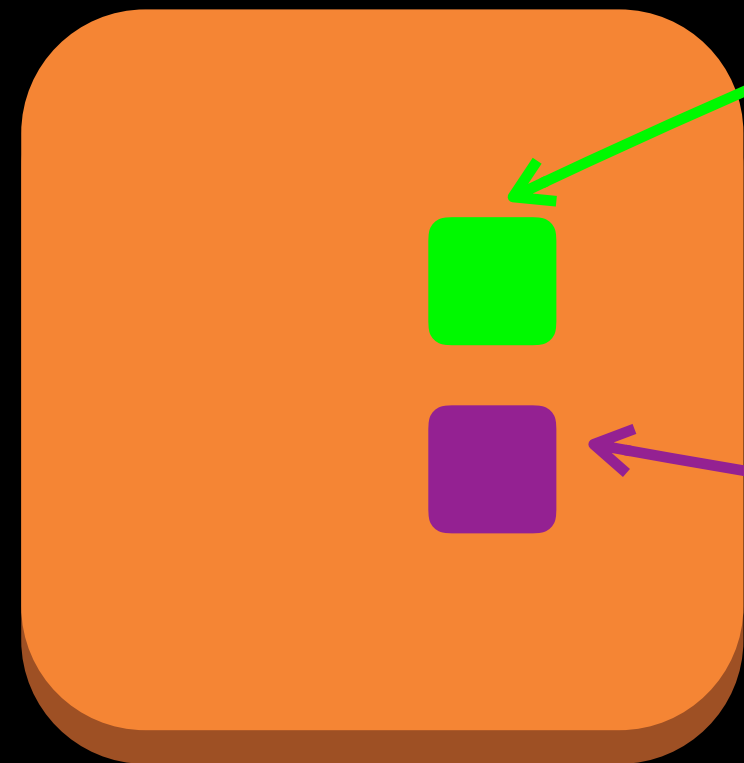
Scheduler

Task Def: WebApp:1
Desired #: 2
Deploy Options:
50% minimum
Load Balancer: X

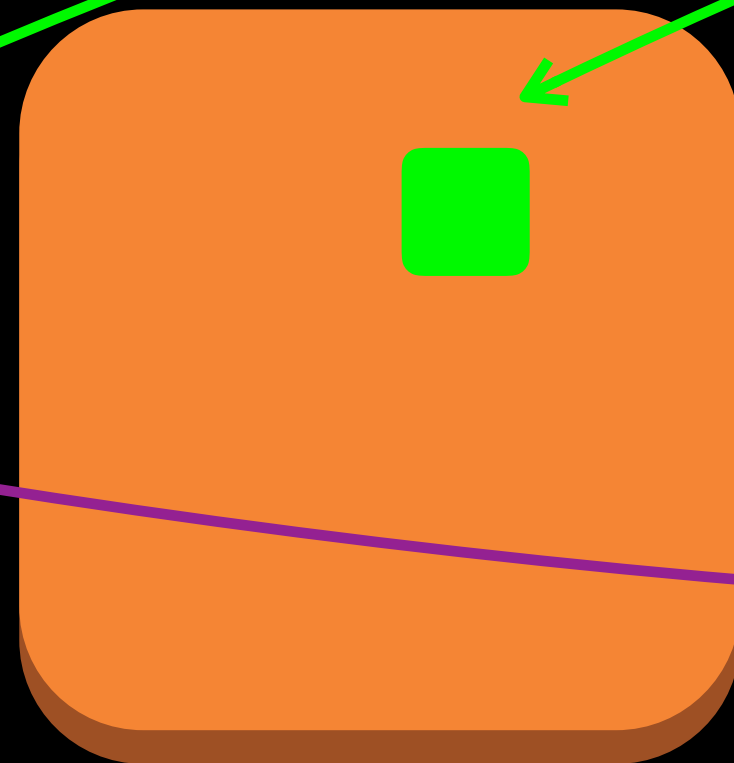
Service: WebApp

Task Def: Cache:1
Desired #: 1
Deploy Options:
50% minimum
Load Balancer: X

Service: Cache



EC2 Instance



EC2 Instance

ECS Cluster

More On Services....

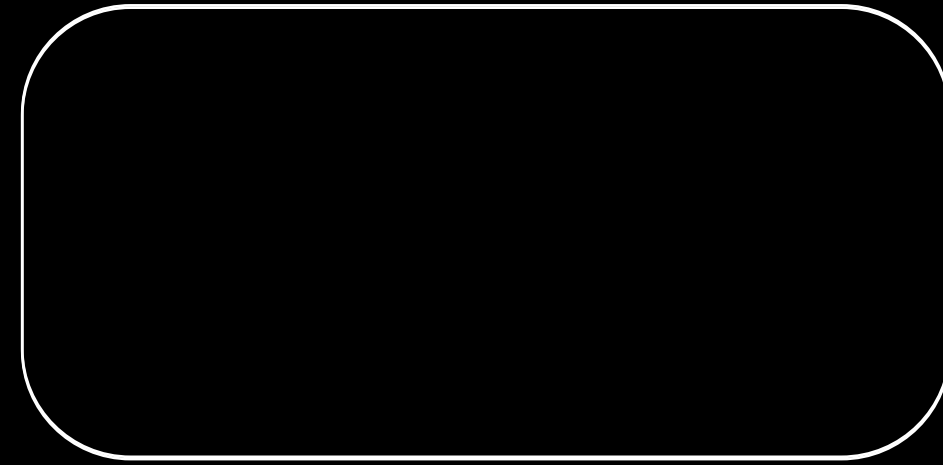
- Continuously attempting to achieve desired count
- Balance system capacity and Availability Zone considerations
- Deployment Strategies
 - Optional
 - Control how many tasks run during the deployment and the ordering of stopping and starting tasks
 - $\text{maximumPercent} - \text{max \# of running tasks} = \text{desiredCount} * \text{maximumPercent}/100$
 - $\text{minimumHealthyPercent} - \text{max \# of running tasks} = \text{desiredCount} * \text{minimumHealthyPercent}/100$
- AutoScaling
 - Scale Service
 - Scale EC2 Instaces

ECS Cluster - Deployment

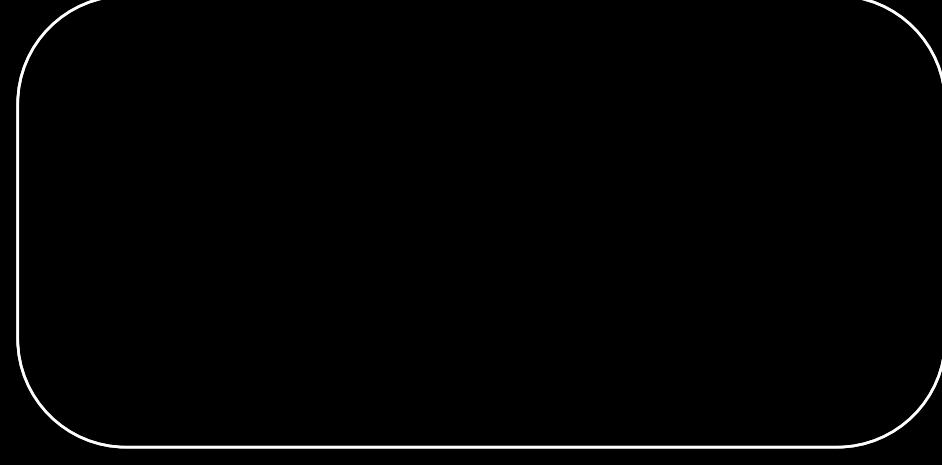
minimumHealthyPercent: 50
maximumPercent: 100



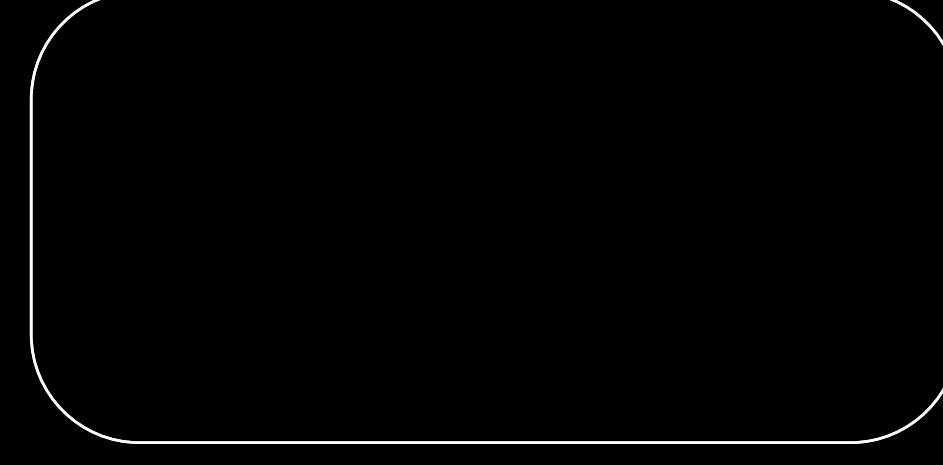
WebApp:2



WebApp:2



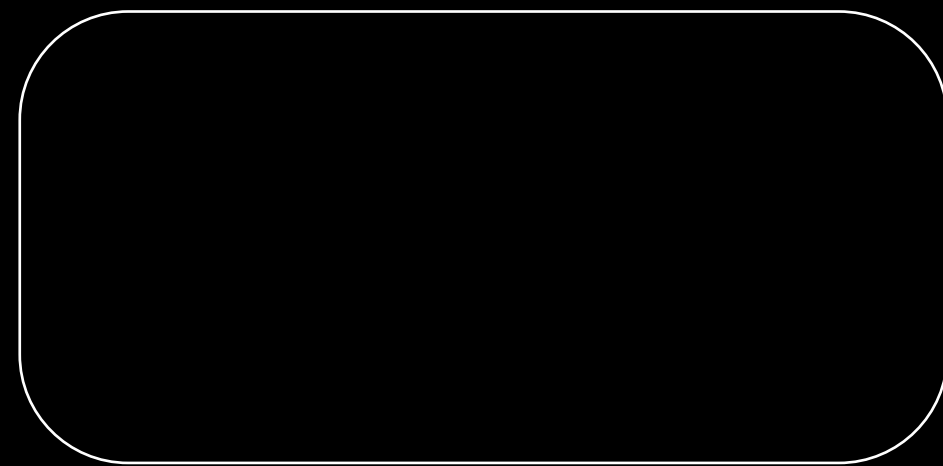
WebApp:2



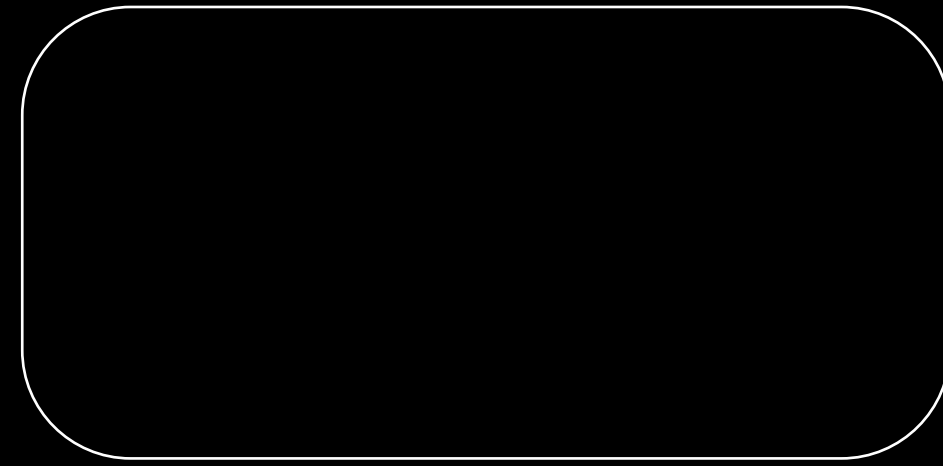
WebApp:2

ECS Cluster - Deployment

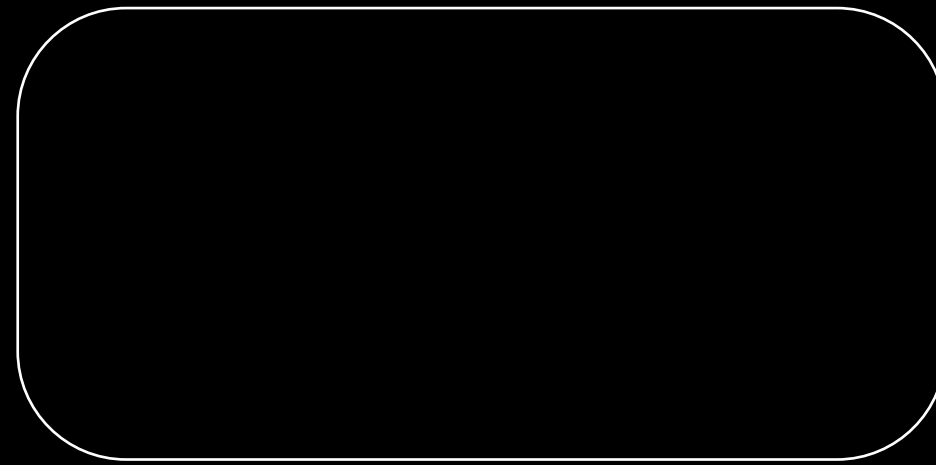
minimumHealthyPercent: 100
maximumPercent: 200



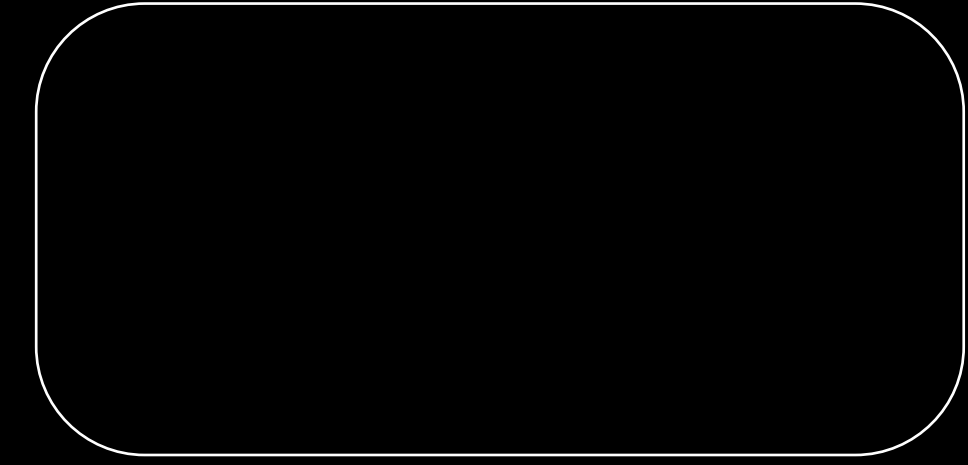
WebApp:1



WebApp:1



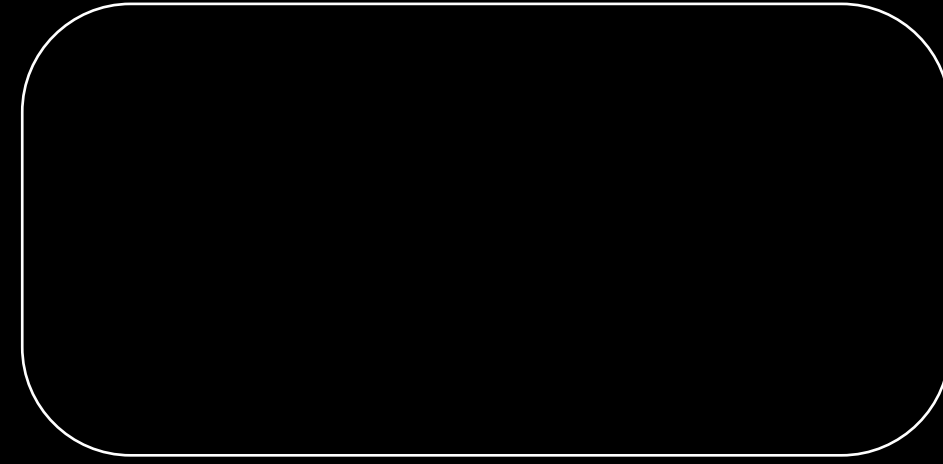
WebApp:1



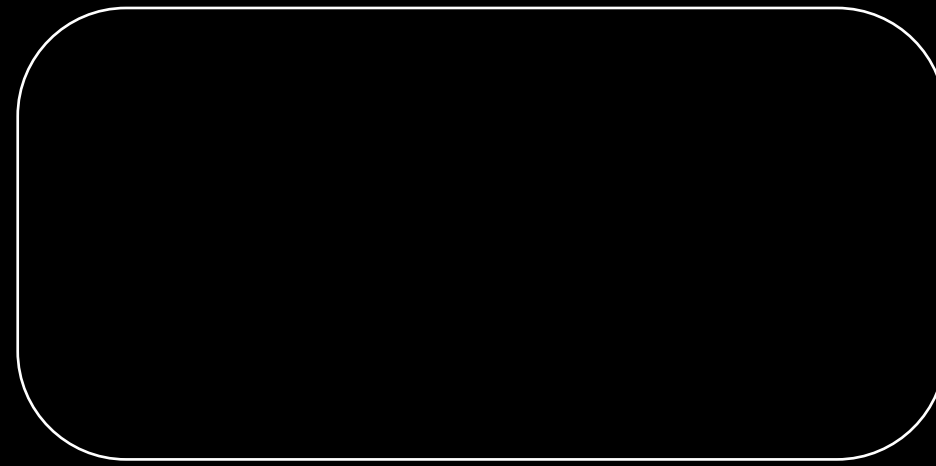
WebApp:1



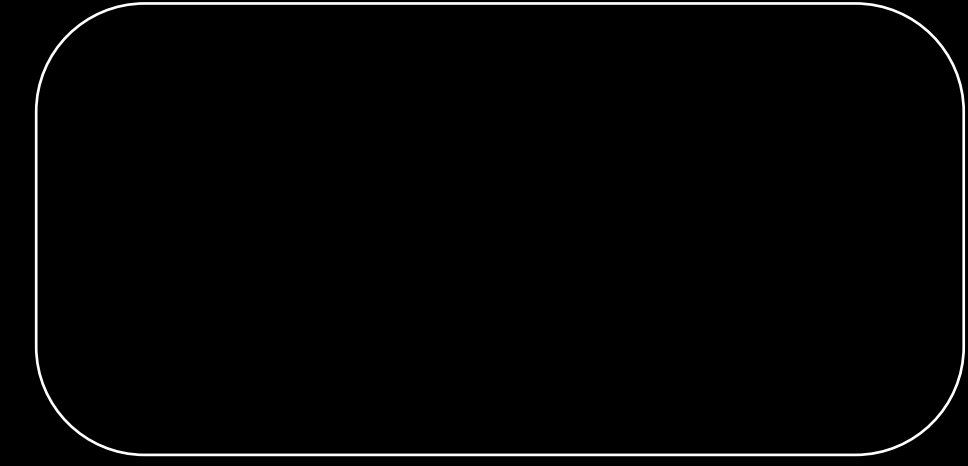
WebApp:2



WebApp:2



WebApp:2



WebApp:2

Even More On Services: Load Balancing

Application ELB
(443) -> Dynamic

Classic ELB
(443) -> 9999

nginx:1.10.1
0:80

nginx:1.10.1
0:80

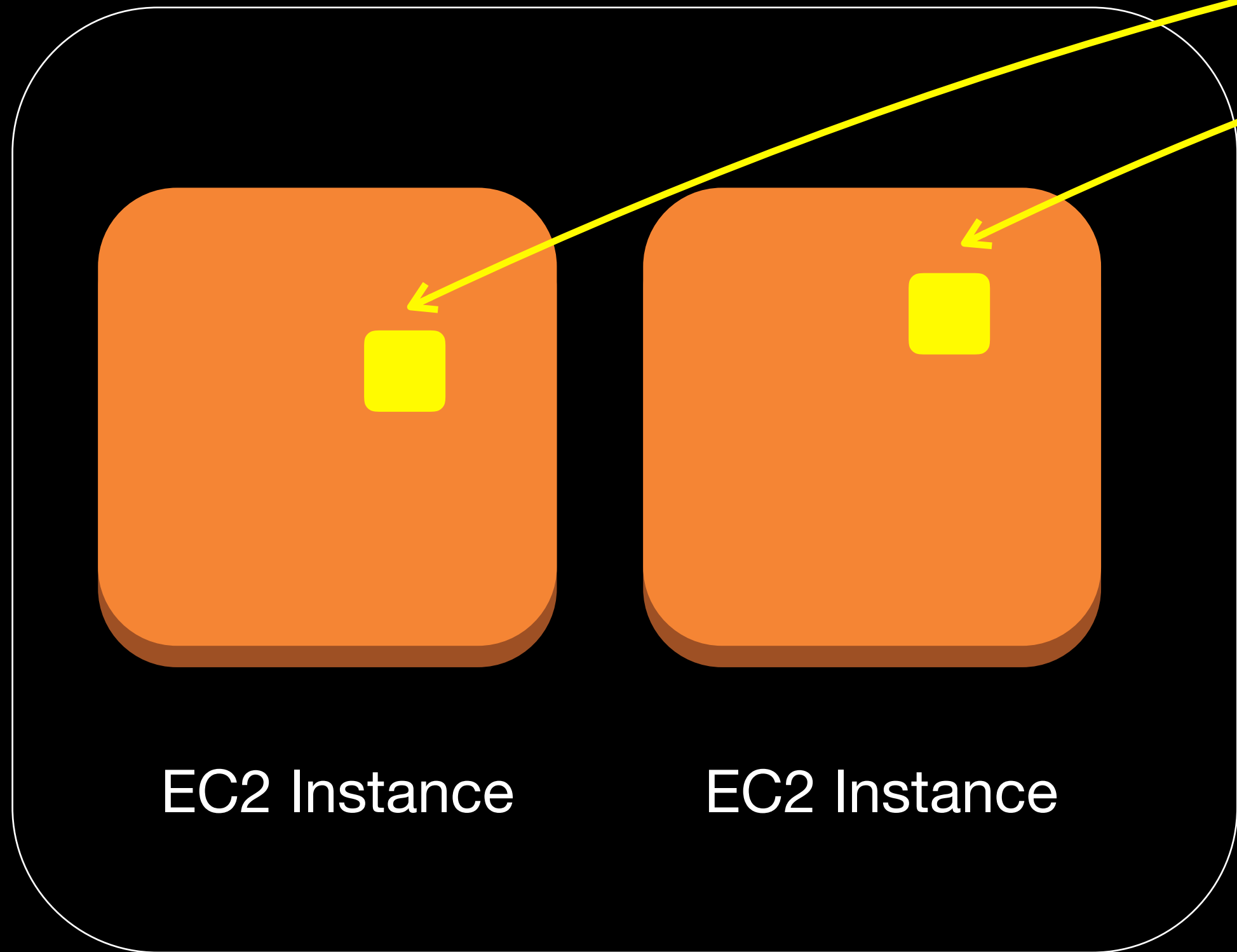
nginx:1.10.1
9999:80

1:1 ELB to Service Ratio!

Scheduling Tasks

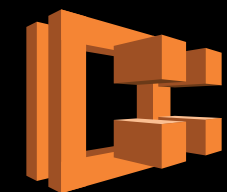


Task Definition: BatchProcessor



ECS Cluster

Randomly distributed via RunTask
or intentionally distributed via
StartTask



Service / Task Definition Design Patterns

ELB

nginx

app

task

EC2 Instance

nginx

app

task

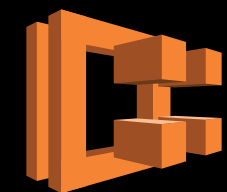
EC2 Instance

nginx

app

task

EC2 Instance



Service / Task Definition Design Patterns

ELB

nginx

app

service

EC2 Instance

nginx

app

service

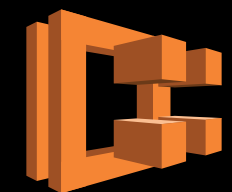
EC2 Instance

nginx

app

service

EC2 Instance



Service / Task Definition Design Patterns

ELB

nginx

dev-app

prod-app

EC2 Instance

nginx

dev-app

prod-app

EC2 Instance

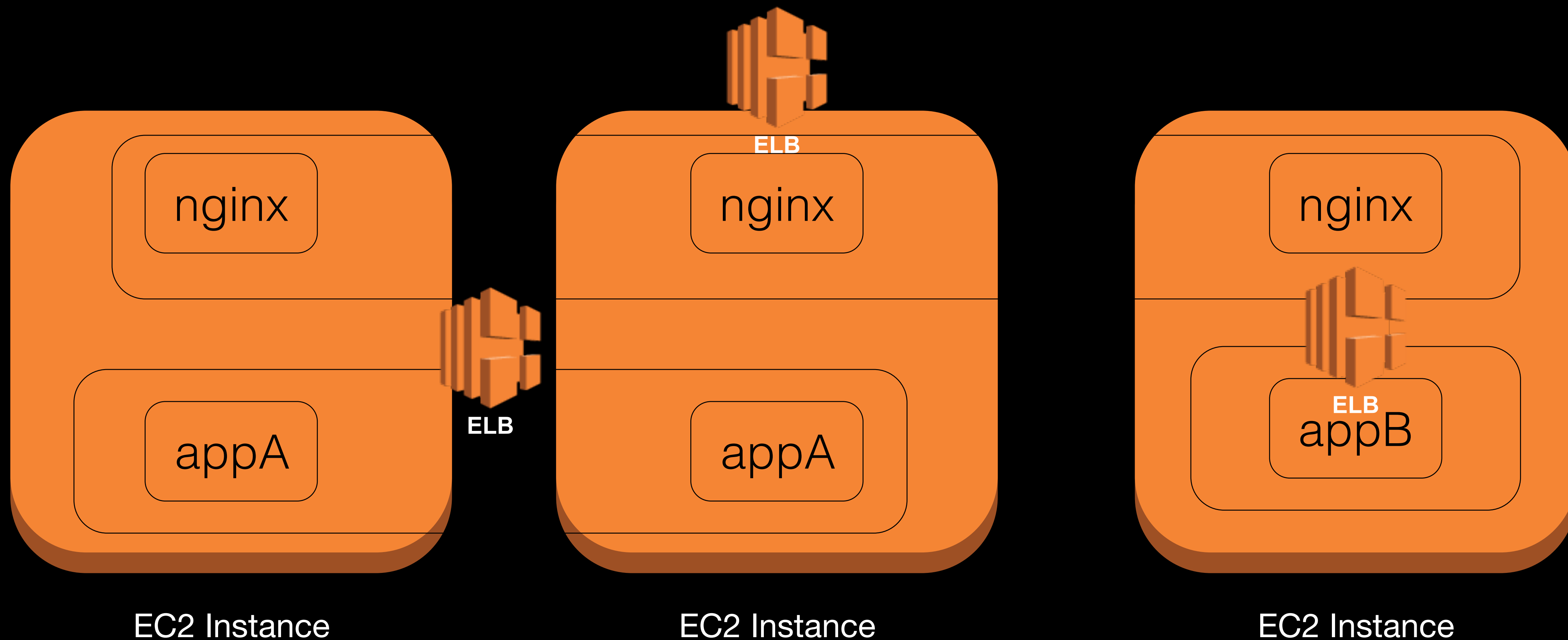
nginx

dev-app

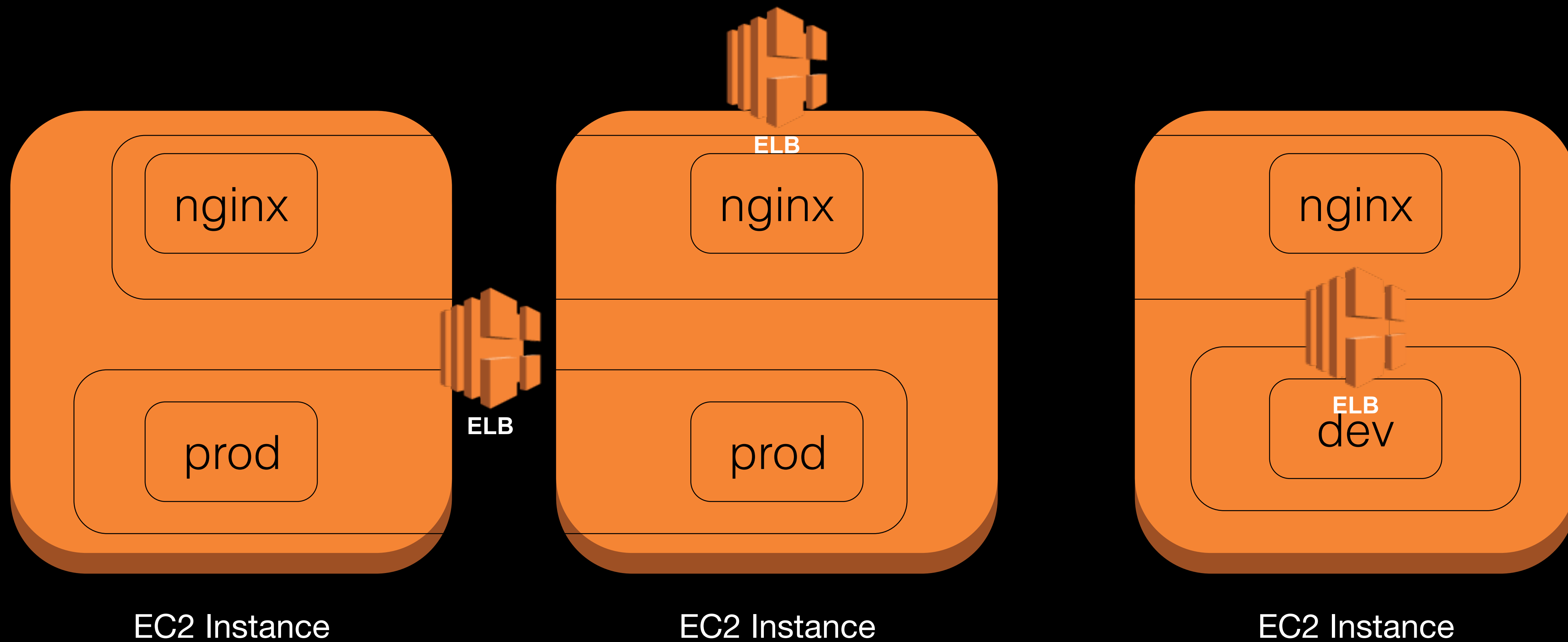
prod-app

EC2 Instance

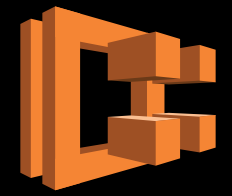
Service / Task Definition Design Patterns



Service / Task Definition Design Patterns



- Container Instance IAM Role (EC2)
 - ecs-agent uses this
- ECS Service Scheduler
 - ECS Service operating on your behalf; ELB register/deregister
- ECS Autoscaling Role
 - Make system adjustments on your behalf
- ***Per task IAM Roles!!!***



- logging w/CloudWatch Logs
- garbage collection
- deploying same task def doesn't do anything (if using same container tag), must create new version
- comparing task definitions
- sharing data: s3? persistent backend? partitioned? EFS?



ECR

- you need to run `aws ecr get-login --region us-east-1 | bash` – which is good for 12 hours
- acs-agent can use instance metadata to pull from ecr, so 3rd party registry credentials can be provided
- There are complaints about the repository naming convention



Amazon
API
Gateway



AWS
Lambda



Amazon
ECS