

# Running Docker clusters on

# AWS

Julien Simon

Principal Technical Evangelist

Amazon Web Services

[julsimon@amazon.com](mailto:julsimon@amazon.com)

[@julsimon](#)

09/11/2016



# The problem

Given a certain amount of processing power and memory,

how can we best manage  
an arbitrary number of apps  
running in Docker containers?



<http://tidalseven.com>

# Docker on Amazon Web Services

## Amazon EC2 Container Service (ECS)

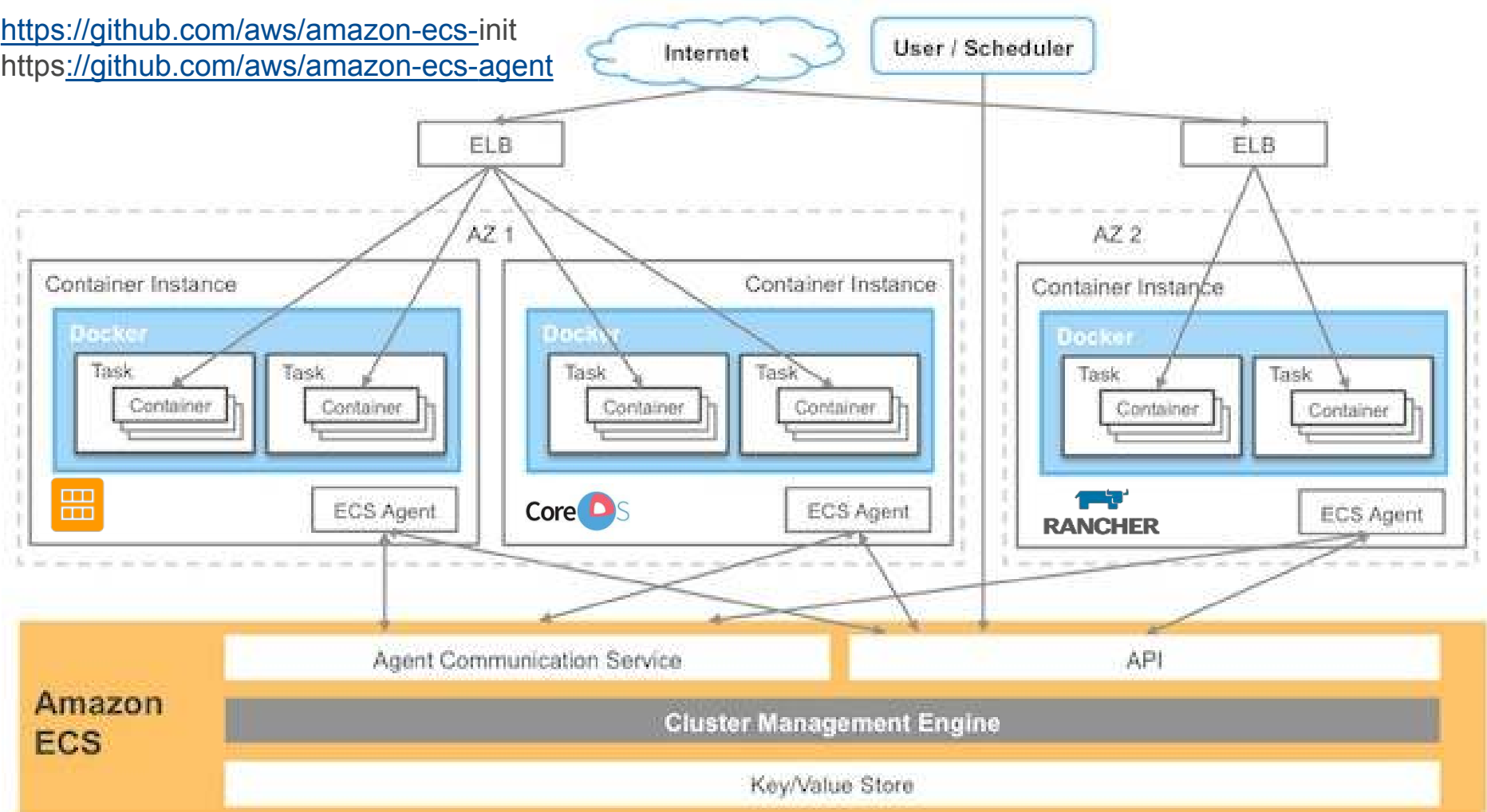
- <https://aws.amazon.com/ecs/>
- Launched in 04/2015
- Available in eu-west-1 & eu-central-1
- No additional charge

## Amazon EC2 Container Registry (ECR)

- <https://aws.amazon.com/ecr/>
- Launched in 12/2015
- Available in eu-west-1 & eu-central-1
- Free tier: 500MB / month for a year
- \$0.10 / GB / month + outgoing traffic



<https://github.com/aws/amazon-ecs-init>  
<https://github.com/aws/amazon-ecs-agent>



Amazon  
ECS

# AWS Partners

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



# Case study: Coursera



<https://www.youtube.com/watch?v=a45J6xAGUvA>

- Coursera deliver Massive Open Online Courses (14 million students, 1000+ courses). Their platform runs a large number of batch jobs, notably to **grade programming assignments**. Grading jobs need to run in **near-real time** while preventing execution of **untrusted code** inside the Coursera platform.
- After trying out some other Docker solutions, Coursera have picked **Amazon ECS** and have even written **their own scheduler**.
- *“Amazon ECS enabled Coursera to focus on releasing new software rather than spending time managing clusters”* - Frank Chen, Software Engineer

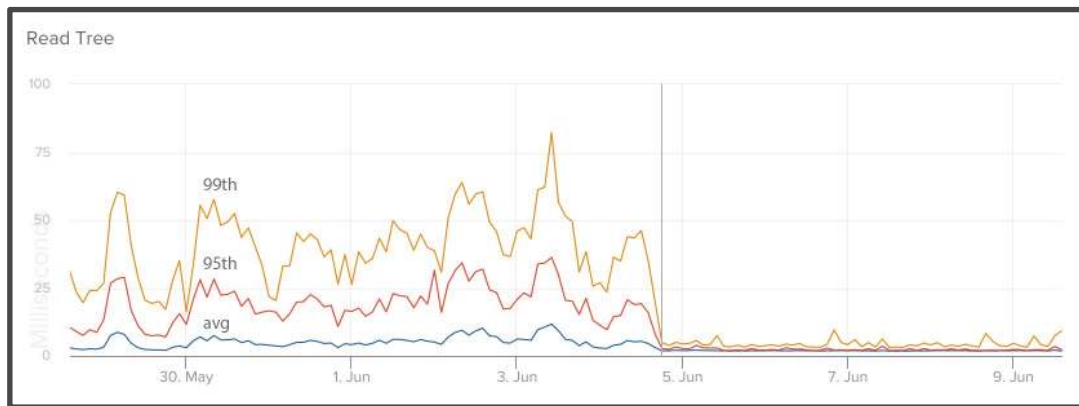


# Case study: Remind



<https://www.youtube.com/watch?v=8zbbQkszP04>

- **Messaging platform** for teachers, parents and students (35M users, 2.5B messages, 50% of U.S. public schools)
- **Micro-service platform** deployed on Heroku, migrated to Amazon ECS (36 nodes in Q4'15)



*“Moving to Amazon ECS significantly improved our service performance”* Jason Fischl, VP of Engineering

# Case study: Segment

# Segment

<https://aws.amazon.com/fr/solutions/case-studies/segment/>

- Segment provides a service used by businesses to collect customer data for later use in **analytics** and **marketing**.
- Different micro-services such as API, CDN, and App are deployed on different **Amazon ECS clusters**. Each service registers to an ELB and Amazon Route 53 points a local entry at each ELB. Services can communicate with each other through DNS.
- Segment have also built **their own PaaS** on top of AWS:  
<https://segment.com/blog/the-segment-aws-stack/>

*“Switching to Amazon ECS has greatly simplified running a service without needing to worry about provisioning or availability”*

Calvin French-Owen, Cofounder and CTO



# The Amazon ECS CLI in one slide

<https://github.com/aws/amazon-ecs-cli>

```
ecs-cli configure --cluster myCluster --region eu-west-1  
ecs-cli up --keypair myKey --capability-iam -size 3  
ecs-cli down myCluster --force
```

```
ecs-cli compose service up  
ecs-cli compose service ps  
ecs-cli compose service scale 8  
ecs-cli compose service stop  
ecs-cli compose service delete
```

```
aws ecs list-clusters  
aws ecs describe-clusters --cluster myCluster  
aws ecs list-container-instances --cluster myCluster
```

Homemade tool: 'ecs-find'

<https://github.com/juliensimon/aws/blob/master/ecs/ecs-find>



# Managing containers on ECS with the console

Services

Tasks

ECS Instances

Metrics

Add additional ECS Instances using [Auto Scaling](#) or [Amazon EC2](#).

Last updated on November 8, 2016 4:31:27 PM (0m ago)



Filter in this page

< Viewing 1-3 Container Instances > Results per page

50



	Container Instance	EC2 Instance	Agent ...	Status	Availa...	Availa...	Agent ...	Docke...
▶	11131f26-49ea-455b-aefc-e69f7963b719	i-0a141a241a...	true	ACTIVE	2048	3764	1.13.0	1.12.1
▶	800093ea-d5d2-456c-a1e7-3e079554460c	i-0b64d3f7fddf...	true	ACTIVE	2048	3764	1.13.0	1.12.1
▶	ca0bb363-4c42-4f16-bc71-8c52a56fb29e	i-0839ee21053...	true	ACTIVE	2048	3764	1.13.0	1.12.1

# Demo #1

## Amazon Linux + Amazon ECS

number of web apps (on port 80) = number of nodes in the cluster

# Managing containers on ECS with Rancher

Hosts

Add Host

ACTIVE



RancherOS1

52.213.1.219 | 1.12.1

RancherOS v0.7.0 (4.4.21-rancher)

2x2.9 GHz | 3.68 GiB | 29.7 GiB

amazonec2

Stack: os

ecs-agent 172.17.0.2

Standalone Containers

+ Add Container

ACTIVE



RancherOS2

52.211.238.190 | 1.12.1

RancherOS v0.7.0 (4.4.21-rancher)

2x2.9 GHz | 3.68 GiB | 29.7 GiB

amazonec2

Stack: os

ecs-agent 172.17.0.2

Standalone Containers

+ Add Container

ACTIVE



RancherOS3

52.213.17.37 | 1.12.1

RancherOS v0.7.0 (4.4.21-rancher)

2x2.9 GHz | 3.68 GiB | 29.7 GiB

amazonec2

Stack: os

ecs-agent 172.17.0.2

Standalone Containers

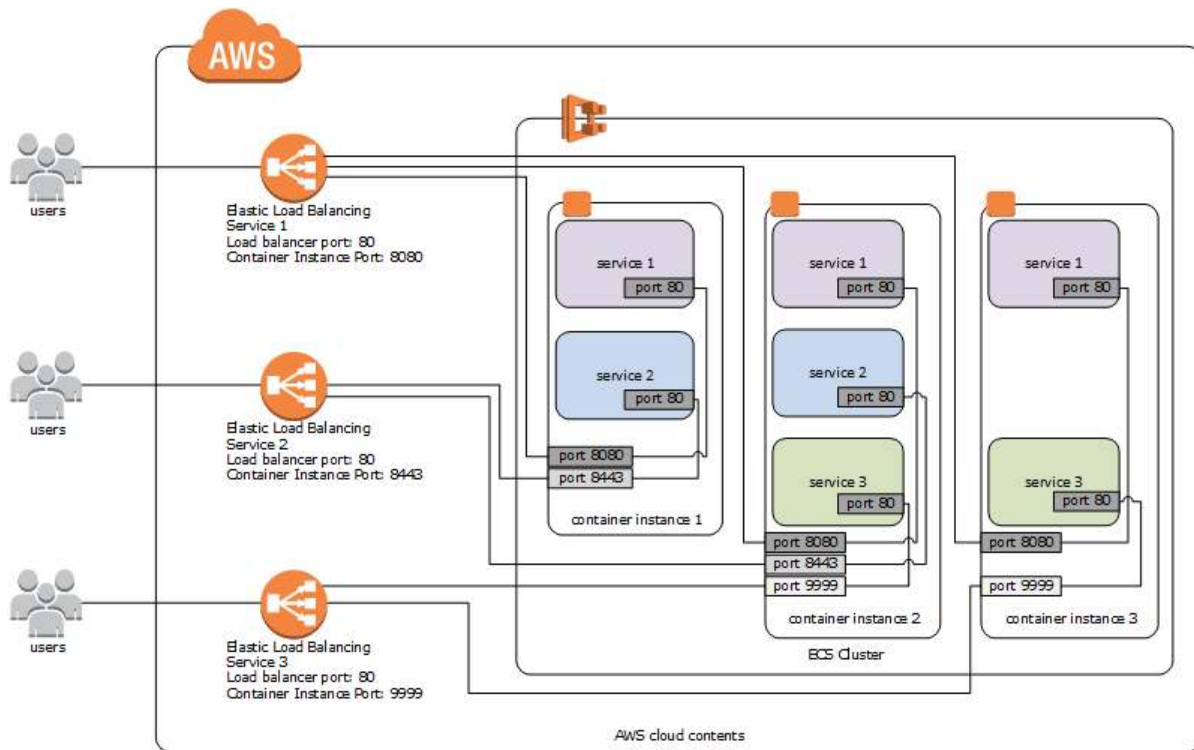
+ Add Container

# Demo #2

## RancherOS + Amazon ECS + Rancher Server

number of web apps (on port 80) = number of nodes in the cluster

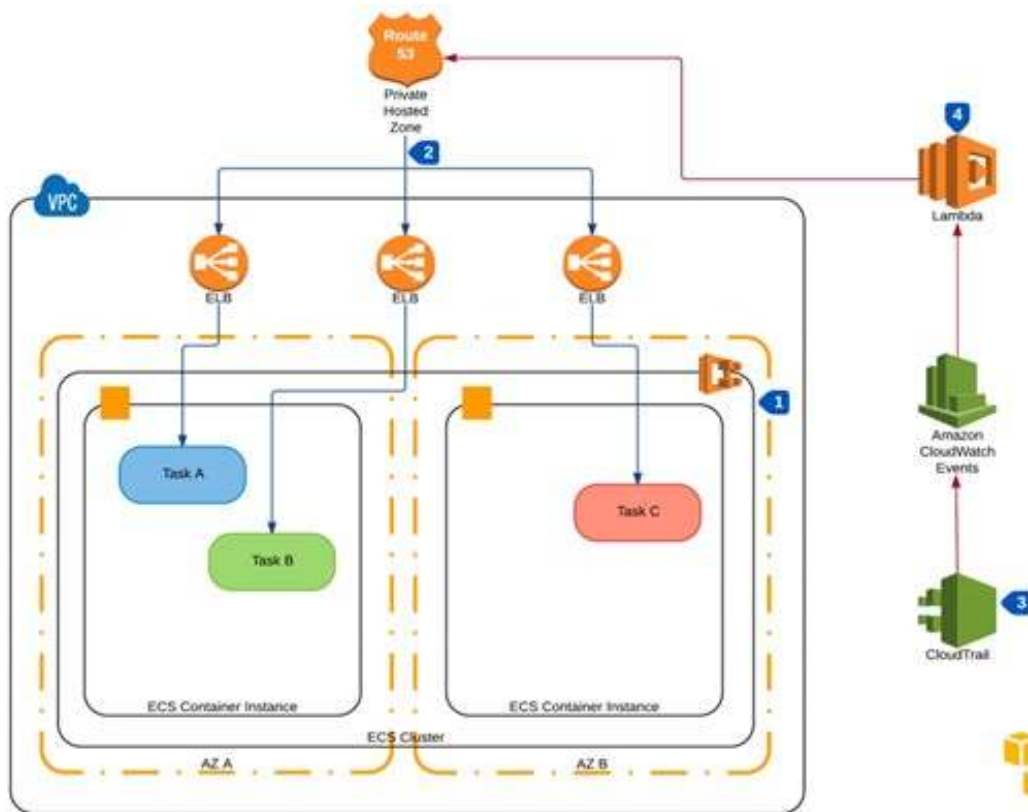
# Architecture #1: Fixed ports + ELB



- No service discovery, no service registration: **automation required**
- Don't use 1 ELB per service. Use 1 single **ALB** instead!
- Only **1 container** from a given image per ECS instance

# Architecture #2: Fixed ports + LB + DNS

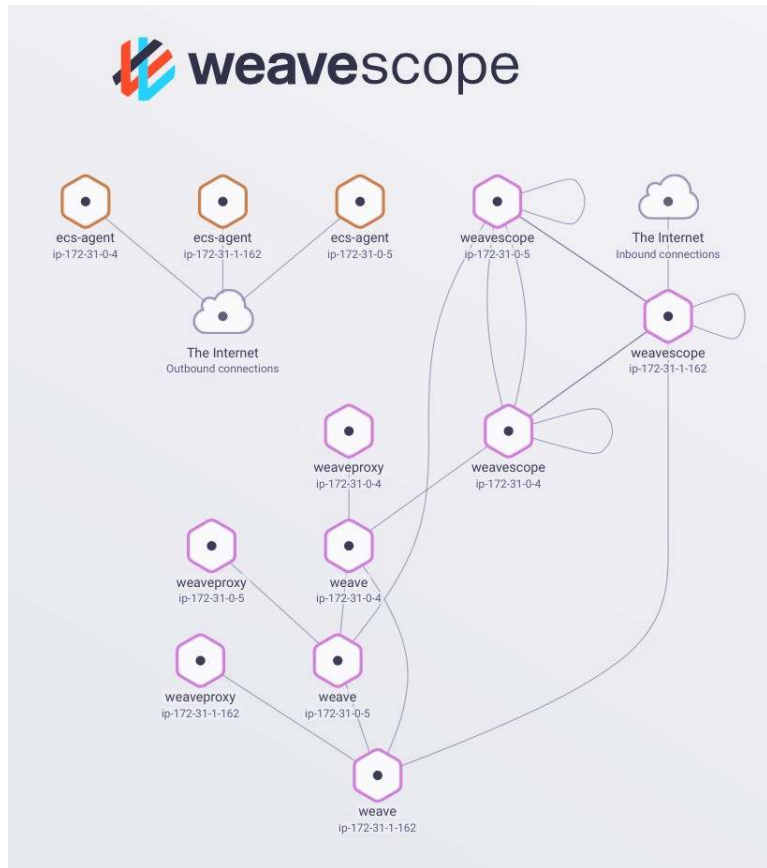
<https://aws.amazon.com/blogs/compute/service-discovery-an-amazon-ecs-reference-architecture/>



- **Service registration** done in Route 53 by CloudWatch Events and Lambda (new CNAME for the LB)
- **Service discovery** with DNS
- Only **1 container** from a given image per ECS instance

# Architecture #3: Fixed ports + Weave + DNS

<https://aws.amazon.com/blogs/apn/architecting-microservices-using-weave-net-and-amazon-ec2-container-service/>



- **Service registration** done by Weave
  - gossip protocol, no central server
  - IP address only (no port)
- **Service discovery & load balancing** done with DNS
- **LB** for Internet-facing services
- Only **1 container** from a given image per ECS instance



# Demo #3

## Amazon Linux + Amazon ECS + Weave

number of web apps (on port 80) = number of nodes in the cluster

<https://aws.amazon.com/blogs/compute/service-discovery-via-consul-with-amazon-ecs/> + tweaks ;)



# Demo #4

## Amazon Linux + Amazon ECS + Consul + Registrator + Fabio

number of web apps (on port 80) > number of nodes in the cluster...

but this is really a complex setup!

### Weather

Add city:

City	Country	Temp
Paris	FR	6.33
Altstadt	DE	2.4
Seattle	US	8.24

Page: **1**

Powered by the [Open Weather Map API](#)

### Stocks

Add stock:

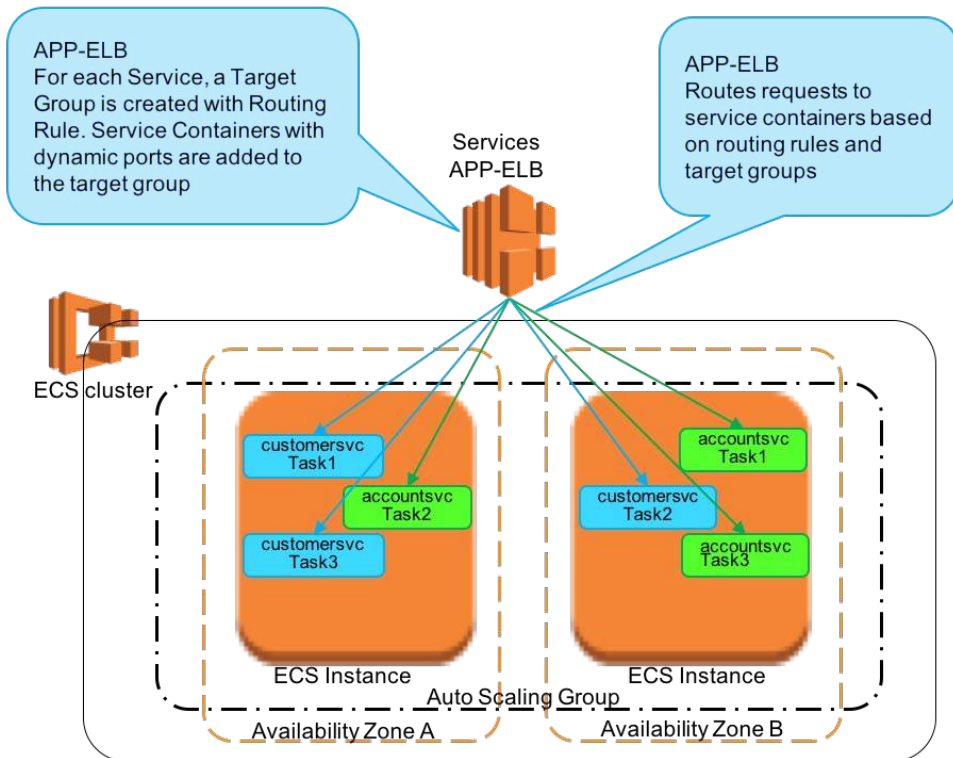
Stock Symbol	Stock Name	Price
amzn	Amazon.com, Inc.	\$781.47
goog	Alphabet Inc.	\$783.63
aapl	Apple Inc.	\$110.03

Page: **1**

Powered by the [Yahoo Finance API](#)

# Architecture #5: Random ports, much simpler!

<https://aws.amazon.com/blogs/compute/microservice-delivery-with-amazon-ecs-and-application-load-balancers/>



- ECS and the ALB work hand in hand for **registration** and **discovery**
- **Multiple containers** from a given image per ECS instance

# Demo #5

## Amazon Linux + Amazon ECS + ALB

number of web apps (on port 80) > number of nodes in the cluster...  
and the setup doesn't make your head hurt (too much)

# Additional resources

Tech articles by Werner Vogels, CTO, Amazon.com

<http://www.allthingsdistributed.com/2014/11/amazon-ec2-container-service.html>

<http://www.allthingsdistributed.com/2015/04/state-management-and-scheduling-with-ecs.html>

<http://www.allthingsdistributed.com/2015/07/under-the-hood-of-the-amazon-ec2-container-service.html>

## Amazon ECS videos @ AWS re:Invent 2015

Amazon ECS: Distributed Applications at Scale <https://www.youtube.com/watch?v=eun8CqGqdk8>

Turbocharge Your Deployment Pipeline with Containers <https://www.youtube.com/watch?v=o4w8opVCI-Q>

From Local Docker Development to Production <https://www.youtube.com/watch?v=7CZFpHUPqXw>

A earlier version of this talk, with a focus on Consul, Registrator & Fabio

<https://www.youtube.com/playlist?list=PLJgojBtbsuc37iqrxro5S5DcMwv63vrTI>

# More sessions

- ~~7/11, 15:00 Hands-on with AWS IoT~~
- ~~8/11, 10:00 A 60-minute tour of AWS Compute~~
- ~~9/11, 10:00 Deep Dive: DevOps on AWS~~
- ~~9/11, 11:00 Running Docker clusters on AWS~~
- 21/11, 11:00 Move fast, build things with AWS
- 22/11, 11:00 Deep Dive: Amazon RDS

# Danke sehr!

Julien Simon  
Principal Technical Evangelist  
Amazon Web Services

julsimon@amazon.com  
@julsimon

