



Amazon ECS

Julien Simon
Technical Evangelist, AWS

Docker Marseille – 16/12/15
Docker Bordeaux – 17/12/2015

The problem

Given an arbitrary amount of processing power and memory (provided by EC2 instances), how can we manage an arbitrary number of apps running in Docker containers in the most efficient and scalable manner ?

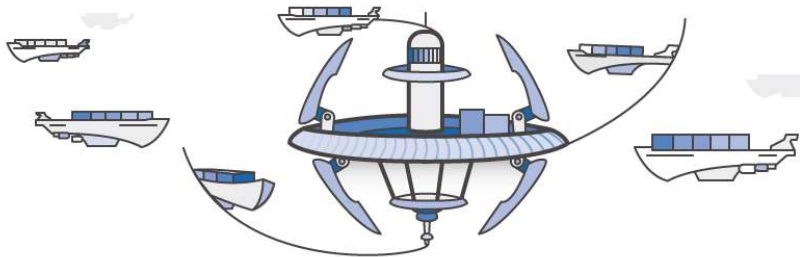


<http://tidalseven.com>

Requirements for modern cluster orchestration

Reliable state management & flexible scheduling

- Cluster must be multi-tenant and scalable
- Cluster must work well for both long-lived and short-lived apps
- Users must be able to focus on writing apps and growing their business
 - High availability must be built-in
 - State management must be built-in



Amazon EC2 Container Service (ECS)

Preview in 11/2014, launch in 04/2015

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

Case study : Coursera

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

What Else Did We Look At?

Home-grown Tech

- Tried, but proved to be unreliable
- Difficult to handle coordination and synchronization



MESOS

- Powerful, but hard to productionize
- Needs developers with experience



kubernetes
a Google

- Designed for GCE first
- Not a managed service, higher Ops load

“Amazon ECS enabled Coursera to focus on releasing new software rather than spending time managing clusters”

Frank Chen, Software Engineer

Case study : Meteor

<https://www.youtube.com/watch?v=xlC3WT6kAVw>

ECS container management

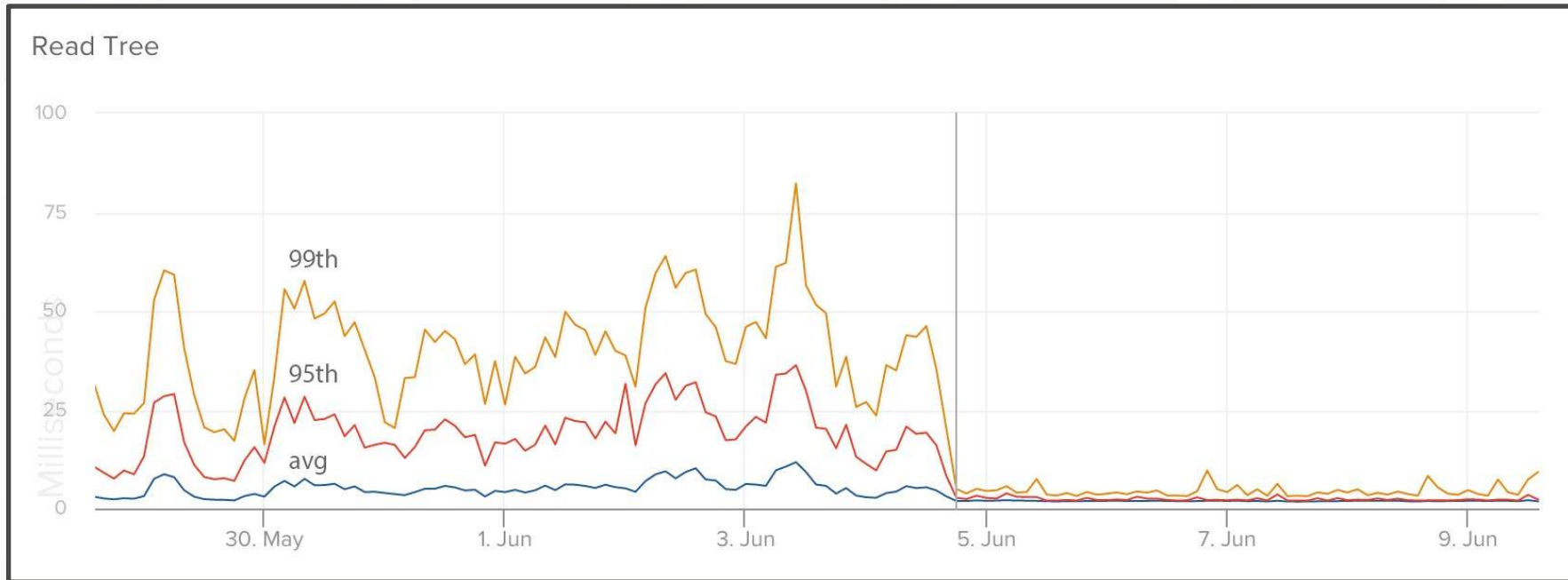
- Lots of exciting options here: ECS, Kubernetes, Marathon, ...
- **Service argument is compelling.** Same case we make for Galaxy to our customers.
- **Integration with other parts of AWS** saves us time and code.
Example: services automatically register containers with Elastic Load Balancing (ELB).
- **Support for multiple Availability Zones.**
- Bottom line: **ECS got us to market faster** than the alternatives.

“ Our big questions were, ‘Can we scale the amount of compute resources necessary to run all our customers’ apps?’ and, ‘Can we scale the mechanics of coordinating all those pieces?’ Using AWS, we can answer ‘yes’ to both”

Matt DeBergalis, Cofounder & VP Product

Case study : Remind

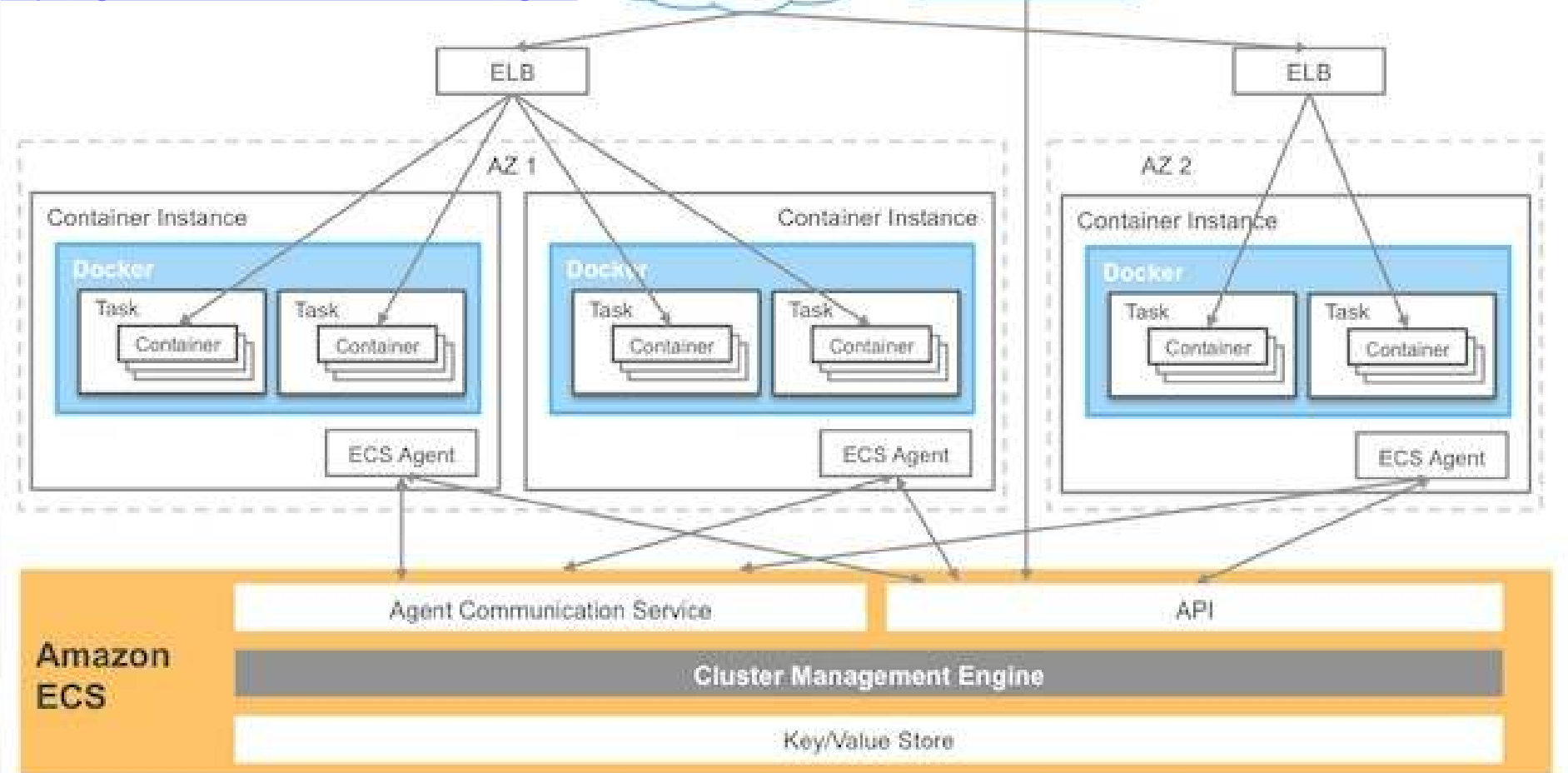
<https://www.youtube.com/watch?v=8zbbQksP04>



*“Moving to Amazon ECS significantly improved our service performance.
We reduced service response times in the 99th percentile by 50%”*

Jason Fischl, VP of Engineering

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



Amazon ECS scalability

100 Node Cluster (End-to-End)					1000 Node Cluster (End-to-End)				
	10%	25%	50%	90%		10%	25%	50%	90%
p50	1.44s	1.44s	1.44s	1.44s	p50	1.29s	1.40s	1.40s	1.41s
p90	1.76s	1.77s	1.85s	2.01s	p90	1.64s	1.63s	1.64s	1.65s
p99	2.05s	2.05s	2.11s	2.21s	p99	1.87s	1.87s	1.87s	1.87s

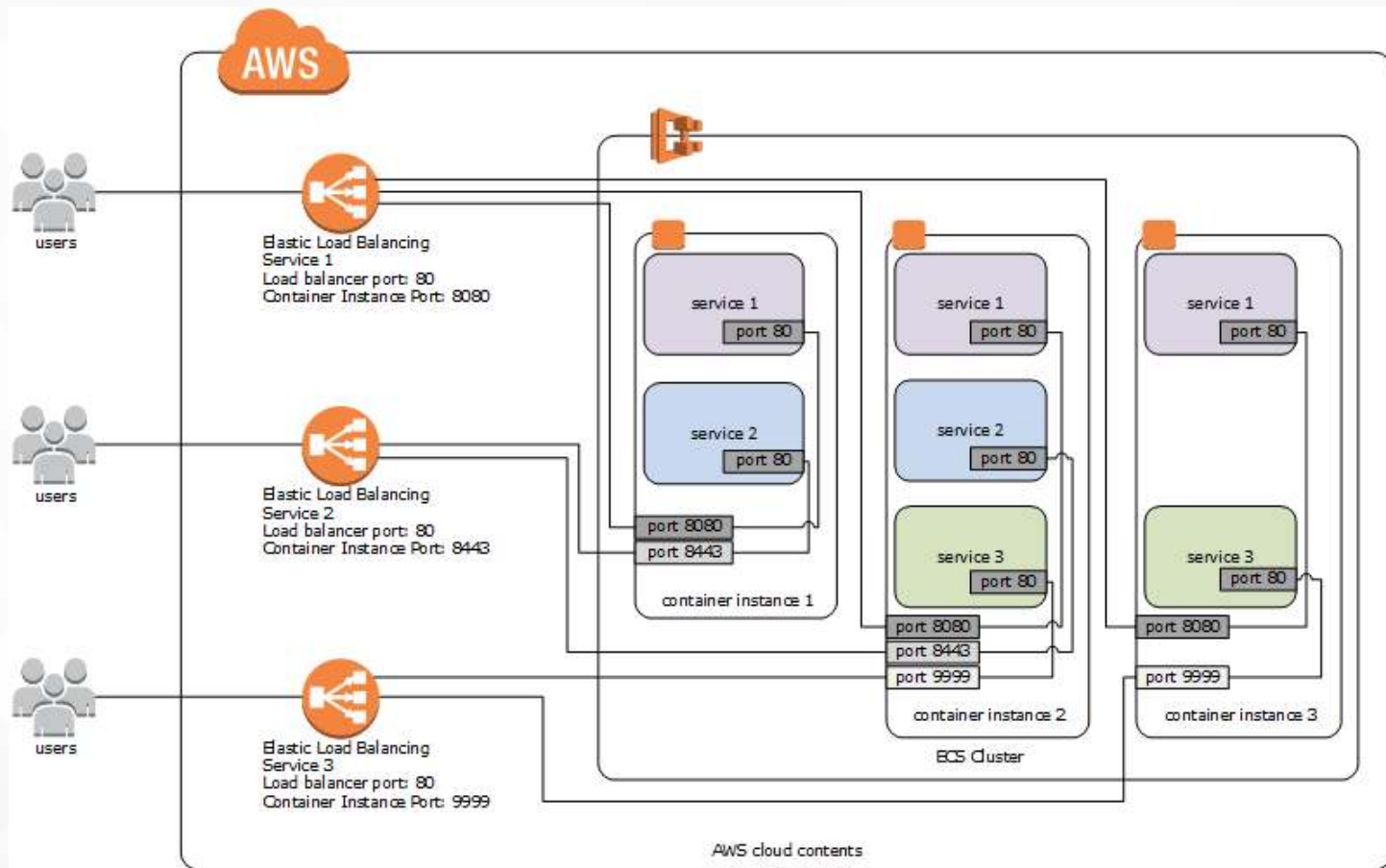
DEMO #1

Demo gods, I'm your humble servant, please be good to me

https://docs.aws.amazon.com/fr_fr/AmazonECS/latest/developerguide/docker-basics.html

<https://github.com/awslabs/ecs-demo-php-simple-app>

Load balancing services on fixed ports



“A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable”

Leslie Lamport, 1987

Micro-services: it just gets worse 😊

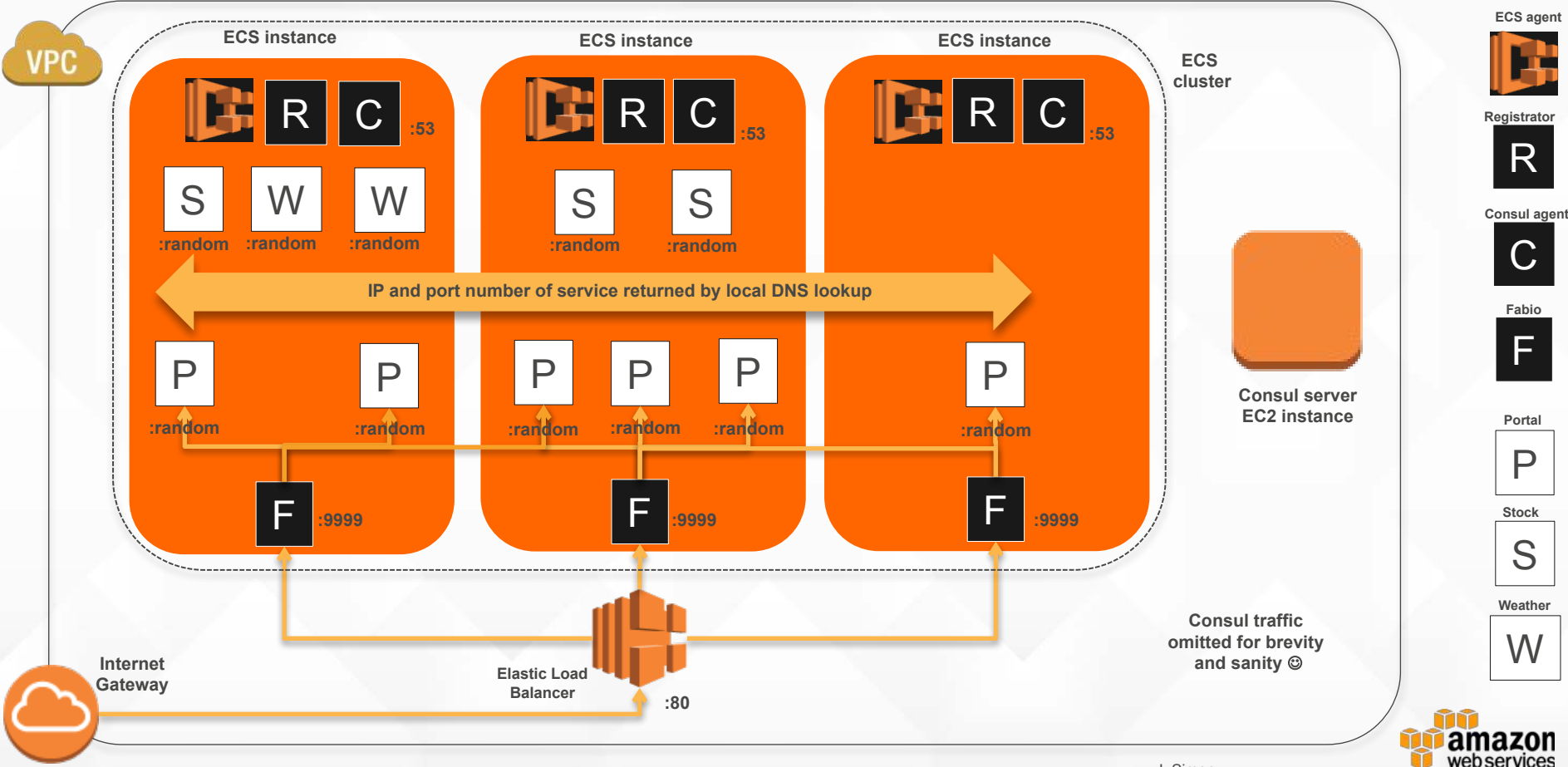
Micro-services run in an ever-moving production environment: continuous deployment, multiple versions running in parallel, servers coming and going, etc.

- Can micro-services be deployed and scaled independently?
- Can multiple copies of a micro-service run on the same server?
- Can micro-services register their name & port automatically?
- Can micro-services discover each other?
- Can traffic be load-balanced across multiple copies of a micro-service?

Yes we can!

- Can micro-services be deployed and scaled independently?
Microservice = Docker container + task definition + service definition
- Can multiple copies of a micro-service run on the same server?
Let Docker assign a random port
- Can micro-services register name & port automatically?
Use Registrator to inspect containers and register them in Consul
- Can micro-services discover each other?
Use local Consul agent for DNS lookups
- Can traffic be load-balanced across multiple copies of a micro-service?
Front → Back : DNS round-robin
User → Front : ELB if running on fixed ports, ELB+Fabio if not

Load balancing services on random ports



DEMO #2

Demo gods, I know I'm pushing it, but please don't let me down now

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

<https://github.com/awslabs/service-discovery-ecs-consul>

<https://www.consul.io/>

<https://github.com/gliderlabs/registrator>

<https://github.com/eBay/fabio>

Thank you! Let's keep in touch 😊

Many events and meetups in 2016

→ [@aws_actus](#) [@julsimon](#)

AWS Summit Paris 2016 : 31/05/2016

Need an AWS speaker?

julsimon@amazon.fr

BONUS SLIDES

Using Amazon ECS

https://docs.aws.amazon.com/fr_fr/AmazonECS/latest/developerguide/ECS_GetStarted.html

AWS Console

<https://console.aws.amazon.com/ecs/>

AWS CLI

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

<https://github.com/aws-labs/aws-shell> **NEW!**

AWS SDK (Java, .NET, Node.js, PHP, Python, Ruby, Go, C++)

https://github.com/aws/aws-sdk-*

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

<https://www.youtube.com/watch?v=MMr78xAiZpQ>

Amazon ECS resources

Tech articles by Werner Vogels, CTO of Amazon

<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>

AWS Compute Blog

<https://aws.amazon.com/blogs/aws/category/ec2-container-service/>

Amazon ECS video @ 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 Deployments <https://www.youtube.com/watch?v=7CZFpHUPqXw>

Creating, scaling and deleting a cluster

```
$ ecs-cli configure -c myCluster -r eu-west-1
```

```
$ ecs-cli up --keypair KEY_PAIR_ID --capability-iam --size 1  
--instance-type t2.micro
```

```
$ ecs-cli scale --size 3 --capability-iam
```

```
$ ecs-cli ps
```

```
$ ecs-cli down myCluster --force
```

Reminder:

- 'ecs-cli up' launches a CloudFormation template
- By default, the cluster is created in a new VPC
- By default, only port 80 is open on ECS instances
- See 'ecs-cli up --help' for advanced networking options

Basic ECS commands

```
$ aws ecs list-clusters
```

```
$ aws ecs describe-clusters --cluster myCluster
```

```
$ aws ecs list-container-instances --cluster myCluster
```

```
$ aws ecs describe-container-instances --cluster myCluster  
--container-instances ECS_INSTANCE_ID
```

Allowing SSH access on an ECS cluster

```
$ aws ecs describe-container-instances --cluster myCluster  
--container-instances ECS_INSTANCE_ID  
--query 'containerInstances[*].ec2InstanceId'
```

```
$ aws ec2 describe-instances --instance-ids EC2_INSTANCE_ID  
--query "Reservations[*].Instances[*].SecurityGroups[*].GroupId"
```

```
$ aws ec2 authorize-security-group-ingress  
--group-id SECURITY_GROUP_ID  
--protocol tcp --port 22 --cidr 0.0.0.0/0
```

ECS metadata

Log into the ECS instance

Instance information

```
$ curl http://localhost:51678/v1/metadata
```

Task information

```
$ curl http://localhost:51678/v1/tasks
```

Creating and scaling an ECS service

Write a **docker-compose.yml** file

\$ ecs-cli compose service **start**

\$ ecs-cli compose service **up**

\$ ecs-cli compose service **scale 3**

Stopping and deleting an ECS service

\$ ecs-cli compose service **stop**

\$ ecs-cli compose service **delete**