



# Contenedores de Amazon desde tu local

Conoce lo que puedes hacer con AWS



# Objetivo

Iniciativa para transferir conocimientos de Amazon AWS desde infraestructura a desarrollo, con el fin de aumentar autonomía, y utilizar/consultar recursos de Oneclick desplegados en la nube de AWS.

# Acceder a AWS

- Consola (<https://oneclickaws.signin.aws.amazon.com/console>): necesita usuario y contraseña
- Línea de comandos: ([https://docs.aws.amazon.com/es\\_es/cli/latest/userguide/cli-chap-install.html](https://docs.aws.amazon.com/es_es/cli/latest/userguide/cli-chap-install.html)): utiliza un par de KEYS para identificar y autorizar llamadas por API.
  - AWS\_ACCESS\_KEY\_ID
  - AWS\_SECRET\_ACCESS\_KEY

Una vez instalado el cliente de línea de comandos, se pueden ejecutar comandos y recibir datos:

```
$ aws --profile oneclick --region eu-central-1 comando subcomando argumentos
```

# Grupos de AWS - Oneclick

## Admins

- R/W Access to all resources
- except Costs and payment methods

## Developers

- R/O Access to all resources
- Approve pipelines
- except costs and payment methods
- Start, Stop, Update ECS Services and Tasks
- Get logs

# Dónde esconde infraestructura las cosas?

Oneclick tiene:

- Antiguamente los recursos estaban en Frankfurt (eu-central-1), Sao Paulo (sa-east-1) (FTD), N. Virginia (us-east-1) (SM y Santillana)
- Ahora todos los recursos están en Frankfurt (eu-central-1)
  - Salvo pai-comercial que sigue en N.Virginia (us-east-1)
- Existen algunas subcuentas: para proyectos de clientes de tipo SaaS
  - gosteam (eu-central-1)
  - santillana-repaso (us-east-1)
  - santillana-wemaths (us-east-1)
  - tangerine (sin definir todavía)

# Clusters compartidos

Para optimización de costes y unificación de procesos de construcción y despliegue, se están unificando los entornos de desarrollo (anteriormente desplegados con un docker-compose manualmente + tareas en Jenkins creadas ad-hoc) utilizando todo lo aprendido del proyecto FTD y de Infraestructura como código.

Clusters AWS Elastic Container Services (ECS)

- Conjunto de EC2 instancias y/o instancias serverless (AWS Fargate) que ejecutan tareas (containers).

Antes una máquina tenía una media de 11 servicios en un proyecto de docker-compose.

Ahora, el cluster de desarrollo ejecuta 70 tareas en tres máquinas, con el coste equivalente de 1 sola\*.

\*Hay costes escondidos, como el balanceador, el sistema de ficheros compartido, el redis..., pero todos se comparten.

# Clusters compartidos

## Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

[Create Cluster](#)[Get Started](#)

View

list

card

3 loaded of 3 clusters

Last updated on November 22, 2021 7:54:02 PM (2m ago)



Filter in this page

Cluster name ▾	CloudWatch monitoring ▾	Services ▾	Running tasks ▾	Pending tasks ▾	Container instances ▾
<a href="#">ecs-oneclick-qa</a>	✔ Default	33	31	0	2
<a href="#">ecs-oneclick-dev</a>	✔ Default	72	70	0	3
<a href="#">ecs-oneclick-prod</a>	✔ Default	17	17	0	1

# Clusters compartidos

Your instance type requirements, budget requirements, and application design will determine how to apply the following best practices for your application. To learn more, see [Spot Instance Best Practices](#).

### Graph

Instance type

Platform

Date range

- ☒ On-Demand price

**\$0.0432** Average hourly cost

☒ ☐ eu-central-1a

**\$0.0148**      Average hourly cost

**65.80%** Average savings

☒ eu-central-1c

**\$0.0141** Average hourly cost

**67.26%** Average savings

☒ ☐ eu-central-1b

**\$0.0143** Average hourly cost

**66.84%** Average savings



# Cómo funciona ECS

**ECS cluster:** conjunto de EC2 instancias y/o instancias serverless (AWS Fargate) que ejecutan tareas (containers).

**Task Definition:** Similar al comando docker run, en el que se le indican las variables de entorno, los volúmenes, los requisitos y los límites del contenedor.

**Service:** Similar a la definición de docker-compose. Mantiene en ejecución un número de tareas ejecutando la versión de Task Definition indicada.

# Otros servicios

- Relational Database Service (RDS): las bases de datos relacionales (mysql)
- ElastiCache: redis
- Elastic Load Balancer (ELB): balanceador de carga
- Simple Storage Service (S3): el almacenamiento de objetos en la nube
- Elastic File System (EFS): un servidor de ficheros NFS para compartir datos entre instancias EC2
- CloudFront: Content Delivery Network (CDN): cache y punto de acceso cercano a clientes
- Route 53: Domain Name Service (DNS): resolución de nombres DNS
- CertificateManager: gestión de certificados SSL (para HTTPS)
- SecretsManager: gestión de credenciales y secretos
- CloudWatch: gestión de logs
- CodePipeline: Pipeline de construcción y despliegue automático
- CodeBuild: Sistema de construcción automático
- Virtual Private Cloud (VPC): Sistema de aislamiento de red
- Elastic Container Registry (ECR): Registro / Repositorio de imágenes docker para desplegar en ECS



# DEMO

## usando WebUI

# Usando AWS CLI

Los comandos devuelven por defecto objetos JSON que pueden ser navegados a través de JSONPath o comando jq:

```
$ aws --profile oneclick-test --region eu-central-1 ecs describe-services --services myproject-dev-auth --cluster myecs-dev
```

# Ejemplo: obtener endpoints de RDS

```
$ aws --profile oneclick-test --region eu-central-1 rds describe-db-clusters | jq '.DBClusters[].Endpoint'
```

```
$ aws --profile oneclick-test --region eu-central-1 rds describe-db-instances | jq  
'DBInstances[].Endpoint.Address'
```

## Ejemplo: obtener un secreto / credencial

```
$ aws --profile oneclick-test --region eu-central-1 secretsmanager list-secrets | jq '.SecretList[].Name'
```

devuelve la lista de secretos definidas en esa region. Se puede filtrar con un `| grep 'project-env'`.

```
$ aws --profile oneclick-test --region eu-central-1 secretsmanager get-secret-value --secret-id  
nombre-del-secreto | jq ".SecretString"
```

devuelve el valor del secreto

## Ejemplo: Navegar por los Builds de un proyecto

```
$ aws --profile oneclick-test --region eu-central-1 codebuild list-projects
```

```
$ aws --profile oneclick-test --region eu-central-1 codebuild list-builds-for-project --project-name  
project-qa-frontoffice
```

```
$ aws --profile oneclick-test --region eu-central-1 logs describe-log-streams --log-group-name  
"/aws/codebuild/project-qa-frontoffice" --order-by LastEventTime --descending | jq -r  
'logStreams[0].logStreamName'
```

```
$ aws --profile oneclick-test --region eu-central-1 logs get-log-events --log-group-name  
"/aws/codebuild/project-qa-frontoffice" --log-stream-name "2ea09e98-bee1-4d8f-b796-ee8f427c28a8" | jq -r  
'events[].message'
```

# Oneclick Oneliners

Comandos de aws cli que permiten obtener información o realizar acciones sin tener que navegar por la consola web de AWS.

Repositorio de scripts en <https://github.com/oneclickdys/ops-tools.git/>





ECS-CLI

# ECS-CLI: acceso a tareas en clusters de ECS

- Es una utilidad reciente de AWS que permite gestionar un cluster de ECS
- De momento para nosotros nos sirve para ver las tareas en ejecución y los logs en desarrollo
- Instalación y configuración:  
[https://docs.aws.amazon.com/es\\_es/AmazonECS/latest/developerguide/ECS\\_CLI\\_installation.html](https://docs.aws.amazon.com/es_es/AmazonECS/latest/developerguide/ECS_CLI_installation.html)

# ECS-CLI: ver tareas en ejecución

```
$ ecs-cli ps --aws-profile oneclick-test --region eu-central-1 --cluster myecs-prod
```

Name	State	Ports	TaskDefinition	Health
myecs-prod/1fb5ca14220c4cefb9bdcf64037cc237/api	RUNNING	18.197.140.28:33541->9501/tcp	project-prod-api:4	UNKNOWN
myeecs-prod/1fdc03c7cf6544afb7d1168314b492cd/frontoffice	RUNNING	18.197.140.28:33546->80/tcp	project-prod-frontoffice:28	UNKNOWN
myecs-prod/28ad94b1f75f4fff969615145249e435/admin-web	RUNNING	18.197.140.28:33410->80/tcp	project-prod-admin-web:2	UNKNOWN

# ECS-CLI: ver logs de una tarea en ejecución

```
$ ecs-cli logs --aws-profile oneclick-test --region eu-central-1 --cluster myecs-prod --task-id 1fb5ca14220c4cefb9bdcf64037cc237
```

```
> ./vendor/bin/phinx migrate -e production
```

```
Phinx by CakePHP - https://phinx.org. 0.10.7
```

```
using config file ./phinx.php
```

```
using config parser php
```

```
using migration paths
```

```
- /usr/src/app/db/migrations
```

```
using seed paths
```

```
using environment production
```

```
using adapter mysql
```



¿PREGUNTAS?