

Implementação de site com AWS ECS



AWS ECS

Orquestre **containers**.
Escale com **facilidade**.
Entregue com **confiança**.

Execução de aplicações containerizadas
com simplicidade, escala e alta disponibilidade.



CONTAINERS

Empacote uma vez,
execute em qualquer lugar.



ESCALABILIDADE

Ajuste automaticamente
à demanda.



CONFIABILIDADE

Alta disponibilidade
e tolerância a falhas.



AWS ECS

Menos infraestrutura.
Mais foco no que importa: sua aplicação.



Amazon ECR
Repositório seguro
de imagens



VPC
Ambiente isolado
e seguro



AWS ECS
Orquestração
de containers
simples e eficiente

Implementação de site com AWS ECS

O que é o ECS

O **Amazon ECS (Elastic Container Service)** é um serviço da Amazon Web Services que permite **executar e gerenciar aplicações em containers** de forma simples e escalável.

Em termos práticos:

- Você empacota sua aplicação em um container (ex: Docker)
- Envia a imagem para um repositório (como o ECR)
- O ECS cuida de **rodar, escalar e manter esses containers funcionando**

De forma bem simples: O ECS é como um **orquestrador de containers**, ou seja, ele decide:

- Onde rodar sua aplicação
- Quantas instâncias precisam estar ativas
- Como manter tudo disponível mesmo com falhas

Implementação de site com AWS ECS

Principais vantagens

- **Escalabilidade automática** conforme a demanda
- **Alta disponibilidade** (distribuí em múltiplas zonas)
- **Integração com outros serviços AWS**
- **Menos esforço operacional** (infra gerenciada)

Resumidamente:

O ECS permite que você foque no código da sua aplicação enquanto a AWS cuida da infraestrutura e da execução dos containers.

Implementação de site com AWS ECS

Proposta



Implementação de site com AWS ECS

Amazon ECR > Private registry > Repositories

Amazon Elastic Container Service

- Private registry
 - Repositories
 - Features & Settings
- Public registry
 - Repositories
 - Settings

ECR public gallery [↗](#)
Amazon ECS [↗](#)
Amazon EKS [↗](#)

Getting started [↗](#)
Documentation [↗](#)

Private repositories (2)

[View push commands](#) [Delete](#) [Actions](#) [Create repository](#)

Q poc X

Repository name ▲	URI	Created at ▼	Tag immutability	Encryption type
No matches No repositories match search filter				

Implementação de site com AWS ECS

Amazon ECR > Private registry > Repositories > Create private repository



Create private repository

General settings

Repository name

Enter a concise name. Repositories support namespaces, which you can use to group similar repositories.

791659331772.dkr.ecr.us-east-2.amazonaws.com/

20 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, and special characters _-./.

Image tag settings [Info](#)

Image tag mutability

Choose the tag mutability setting.

- Mutable**
Image tags can be overwritten.
- Immutable**
Image tags can't be overwritten.

Mutable tag exclusions

Tags that match these filters will be immutable (can't be overwritten). Using wildcards (*) will match zero or more image tag characters.

Add filter

Filters must only contain letters, numbers, and special characters (._*-). Each filter is limited to 128 characters, 2 wildcards (*), and you can add up to 5 filters in the exclusion list.

Implementação de site com AWS ECS

Amazon ECR > Private registry > Repositories

Amazon Elastic Container Service

- Private registry
 - Repositories
 - Features & Settings
- Public registry
 - Repositories
 - Settings

ECR public gallery [↗](#)
Amazon ECS [↗](#)
Amazon EKS [↗](#)

Getting started [↗](#)
Documentation [↗](#)

Successfully created private repository, culturadevops-poc-01

Private repositories (3)

View push commands Delete Actions Create repository

Q poc

Repository name	URI	Created at	Tag immutability	Encryption type
culturadevops-poc-01	791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01	April 14, 2026, 14:30:51 (UTC-03)	Mutable	AES-256

Implementação de site com AWS ECS

Push commands for culturadevops-poc-01



macOS / Linux

Windows

Make sure that you have the latest version of the AWS TOOLS for PowerShell and Docker installed. For more information, see [Getting Started with Amazon ECR](#).

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS TOOLS for PowerShell:

```
(Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin 791659331772.dkr.ecr.us-east-2.amazonaws.com
```

Note: If you receive an error using the AWS TOOLS for PowerShell, make sure that you have the latest version of the AWS TOOLS for PowerShell and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
docker build -t culturadevops-poc-01 .
```

3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag culturadevops-poc-01:latest 791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01:latest
```

4. Run the following command to push this image to your newly created AWS repository:

```
docker push 791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01:latest
```

Close

Implementação de site com AWS ECS

> Disco Local (C:) > Fontes > culturadevops > ECS

Nome	Data de modificação	Tipo	Tamanho
site	14/04/2026 15:39	Pasta de arquivos	
build-ecr.bat	14/04/2026 15:35	Arquivo em Lotes ...	1 KB
Dockerfile	10/12/2024 13:23	Arquivo	1 KB

> Disco Local (C:) > Fontes > culturadevops > ECS > site >

Nome	Data de modificação	Tipo	Tamanho
assets	14/04/2026 15:39	Pasta de arquivos	
index.html	10/12/2024 13:03	Chrome HTML Do...	26 KB
readme.txt	10/12/2024 13:03	Documento de Te...	1 KB

Implementação de site com AWS ECS

Dockerfile - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
from nginx
copy ./site /usr/share/nginx/html
Expose 80
```

build-ecr.bat - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 791659331772.dkr.ecr.us-east-2.amazonaws.com
docker build -t culturadevops-poc-01 .
docker tag culturadevops-poc-01:latest 791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01:latest
docker push 791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01:latest
```

Implementação de site com AWS ECS

```
ECS > dir

Diretório: C:\Fontes\culturadevops\ECS

Mode                LastWriteTime         Length Name
----                -
d-----          14/04/2026   15:39             site
-a-----          14/04/2026   15:35           376 build-ecr.bat
-a-----          10/12/2024   13:23           56 Dockerfile

ECS > ..\..\futura\utilitarios\futura-utilitarios\assume-role\windows\Credencial.ps1
=====
AWS Login (PowerShell)
=====
Access Key ID:
```

```
Autenticado com sucesso.

Credenciais expiram em: 04/14/2026 17:49:15

Pronto para uso com AWS CLI / Terraform.
Rogerio P. Calixto > ECS > ✓
```

Implementação de site com AWS ECS

```
ECS dir

Diretório: C:\Fontes\culturadevops\ECS

Mode                LastWriteTime         Length Name
----                -
d-----          14/04/2026   15:39             site
-a----          14/04/2026   15:35           376 build-ecr.bat
-a----          10/12/2024   13:23           56 Dockerfile

Rogério P. Calixto ECS .\build-ecr
```

Implementação de site com AWS ECS

```
ECS .\build-ecr

C:\Fontes\culturaredevops\ECS>aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 791659331772.dkr.ecr.us-east-2.amazonaws.com
Login Succeeded

C:\Fontes\culturaredevops\ECS>docker build -t culturaaredevops-poc-01 .
[*] Building 10.8s (8/8) FINISHED                                docker:default
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                    0.0s
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 93B                                0.0s
=> [internal] load metadata for docker.io/library/nginx:latest  2.0s
=> [auth] library/nginx:pull token for registry-1.docker.io    0.0s
=> [internal] load build context                                 0.3s
=> => transferring context: 3.58MB                                0.0s
=> [1/2] FROM docker.io/library/nginx@sha256:7f0adca1fc6c29c8dc49a2e99037a10ba20dc266baaed0988e9fb4d0d8b85ba0  8.0s
=> resolve docker.io/library/nginx@sha256:7f0adca1fc6c29c8dc49a2e99037a10ba20dc266baaed0988e9fb4d0d8b85ba0  0.0s
=> sha256:5435b2dcdcf5cb7faa0d5b1d4d54be2c72a776fab9a605336f5067d6e9ecb5976  29.78MB / 29.78MB
=> sha256:88d1d984b765ca06bdffb2c450ede950034501dad79536244e8fcd4f1444840a  628B / 628B
=> sha256:7f0adca1fc6c29c8dc49a2e99037a10ba20dc266baaed0988e9fb4d0d8b85ba0  10.23kB / 10.23kB
=> sha256:a716c9c12c382ab51a71127f1dd9440a6f118939b92af2814d4f6232bb6105d4  9.09kB / 9.09kB
=> sha256:f14ce3095f46ab65fd053991588a243e2d7b45f6d826c93d69c37d4307350  2.29kB / 2.29kB
=> sha256:054715a6bffa715b31d05aa5cf6aac8423bd97a1981d1d69a1a71d530ce588e6  33.16MB / 33.16MB
=> sha256:4a038fd18db12b39452e6f5f883577e987b3ff96e8e5537079119e21086f28b  955B / 955B
=> sha256:84e114c2bb367b07ccb9aff4dbcc37d7a8f119884219f2efc2cfd702a8510b9f4  404B / 404B
=> sha256:7b5d674621c2c637ede5eb94b8a1a844d849231ae61df78fc31315ce35e4bf  1.21kB / 1.21kB
=> sha256:448ea5cac5d5181193a0d6e6186ea1673e3713f929b4bb911ad63d2a6ef6155f  1.40kB / 1.40kB
=> extracting sha256:5435b2dcdcf5cb7faa0d5b1d4d54be2c72a776fab9a605336f5067d6e9ecb5976  2.2s
=> extracting sha256:054715a6bffa715b31d05aa5cf6aac8423bd97a1981d1d69a1a71d530ce588e6  0.0s
=> extracting sha256:88d1d984b765ca06bdffb2c450ede950034501dad79536244e8fcd4f1444840a  0.0s
=> extracting sha256:4a038fd18db12b39452e6f5f883577e987b3ff96e8e5537079119e21086f28b  0.0s
=> extracting sha256:84e114c2bb367b07ccb9aff4dbcc37d7a8f119884219f2efc2cfd702a8510b9f4  0.0s
=> extracting sha256:7b5d674621c2c637ede5eb94b8a1a844d849231ae61df78fc31315ce35e4bf  0.0s
=> extracting sha256:448ea5cac5d5181193a0d6e6186ea1673e3713f929b4bb911ad63d2a6ef6155f  0.0s
=> [2/2] COPY ./site /usr/share/nginx/html
=> exporting image                                             0.0s
=> exporting layers                                           0.0s
=> writing image sha256:1ac5f48b4bd3588a69682c164b7a9795836af4dd57b2f368b845a95aa6e6cf2b  0.0s
=> naming to docker.io/library/culturaaredevops-poc-01       0.0s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

C:\Fontes\culturaredevops\ECS>docker tag culturaaredevops-poc-01:latest 791659331772.dkr.ecr.us-east-2.amazonaws.com/culturaaredevops-poc-01:latest

C:\Fontes\culturaredevops\ECS>docker push 791659331772.dkr.ecr.us-east-2.amazonaws.com/culturaaredevops-poc-01:latest
The push refers to repository [791659331772.dkr.ecr.us-east-2.amazonaws.com/culturaaredevops-poc-01]
04afac2590c7: Pushed
0d09dfdcffeb: Pushed
5ca2f881e5bf: Pushed
db010d6b6a3c: Pushed
3fd3ceafa064: Pushed
4ef7ac306afb: Pushed
4db53caabf0: Pushed
68c70d0dd9aa: Pushed
latest: digest: sha256:784d159bcaa446c571a38272633e12631f83b7d6de3a268dcf7ea37d5c2f1686 size: 1989
```

Implementação de site com AWS ECS

Amazon ECR > Private registry > Repositories > culturadevops-poc-01

Amazon Elastic Container Service

- Private registry
 - Repositories
 - culturadevops-poc-01**
 - Features & Settings
 - Public registry
 - Repositories
 - Settings
- ECR public gallery
- Amazon ECS
- Amazon EKS

culturadevops-poc-01

Summary | **Images** | Lifecycle policy | Permissions | Repository tags

Images (1) Info 🔄 Delete 📄 Copy URI Details Scan View push commands

🔍 Filter active images

<input type="checkbox"/>	Image tags	Type	Created at	Image size (MB)	Image digest	Last pulled at
<input type="checkbox"/>	latest	Image	April 14, 2026, 15:57:42 (UTC-03)	66.65	📄 sha256:784d159b...	-

Implementação de site com AWS ECS

Amazon ECR > Private registry > Repositories > culturadevops-poc-01

Amazon Elastic Container Service

Private registry

- Repositories
 - culturadevops-poc-01
- Features & Settings

Public registry

- Repositories
- Settings

ECR public gallery [↗](#)

Amazon ECS [↗](#)

Amazon EKS [↗](#)

culturadevops-poc-01

Summary | **Images** | Lifecycle policy | Permissions | Repository tags

Image URI copied

Images (1) Info Refresh Delete Copy URI Details Scan View push commands

Filter active images

<input checked="" type="checkbox"/>	Image tags	Type	Created at	Image size (MB)	Image digest	Last pulled at
<input checked="" type="checkbox"/>	latest	Image	April 14, 2026, 15:57:42 (UTC-03)	66.65	sha256:784d159b...	-

```
791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01:latest
```

Implementação de site com AWS ECS

Amazon Elastic Container Service > Task definitions

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions [New](#)
- Account settings

- [AWS Batch](#)
- [Amazon ECR](#)
- [Repositories](#)

Task definitions (3) [Info](#)

Last updated April 14, 2026, 16:28 (UTC-3:00) [Refresh](#) [Deploy](#) [Create new revision](#) [Create new task definition](#)

Filter task definitions: No matches [Clear filters](#) Filter status: Active

Task definition	Status of last revision
<p>No task definitions No task definitions to display.</p> Create new task definition	

Implementação de site com AWS ECS

Amazon Elastic Container Service > Task definitions

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions [New](#)
- Account settings

AWS Batch [↗](#)

Amazon ECR [↗](#)

Task definitions (3) [Info](#)

Last updated April 14, 2026, 16:28 (UTC-3:00) [↻](#) [Deploy](#) [Create new revision](#) [Create new task definition](#)

No matches [Filter status](#) Active

[poc](#) [X](#) [Clear filters](#)

[Create new task definition](#)

[Create new task definition with JSON](#)

< 1 > [⚙️](#)

Task definition	Status of last revision
-----------------	-------------------------

No task definitions
No task definitions to display.

[Create new task definition](#)

Implementação de site com AWS ECS

The screenshot shows the AWS Management Console interface for creating a new task definition. The left sidebar contains navigation options for Amazon Elastic Container Service, including Express Mode, Clusters, Namespaces, Task definitions (highlighted), Daemon task definitions, and Account settings. The main content area is titled 'Create new task definition' and includes sections for configuration, infrastructure requirements, and launch type.

Amazon Elastic Container Service > Create new task definition

Create new task definition [Info](#)

Task definition configuration

Task definition family | [Info](#)
Specify a unique task definition family name.

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

▼ Infrastructure requirements [Info](#)

Specify the infrastructure requirements for the task definition.

Launch type | [Info](#)
Selection of the launch type will change task definition parameters.

- AWS Fargate**
Serverless compute for containers.
- Managed Instances**
Use if you have specific hardware constraints, such as GPU accelerators, CPU instruction sets, or network-optimized hardware (offloads scaling, patching, and instance management to AWS).
- Amazon EC2 instances**
Self-managed infrastructure using Amazon EC2 instances.

OS, Architecture, Network mode
Network mode is used for tasks and is dependent on the compute type selected.

Implementação de site com AWS ECS

The screenshot shows the AWS Elastic Container Service (ECS) console interface for creating a new task definition. The page is titled "Create new task definition" and features a left-hand navigation menu with options like "Express Mode", "Clusters", "Namespaces", "Task definitions", "Daemon task definitions", and "Account settings".

The main configuration area includes the following sections:

- Operating system/Architecture:** Set to "Linux/X86_64".
- Network mode:** Set to "awsvpc".
- Task size:** A dropdown menu with the text "Specify the amount of CPU and memory to reserve for your task." and the value ".5 vCPU".
- Memory:** A dropdown menu with the value "1 GB".
- Task roles - conditional:** A section with a "Task role" dropdown set to "-" and a "Create new role" button.
- Task execution role:** A section with a dropdown set to "ecsTaskExecutionRole" and a "Create new role" button.
- Task placement - optional:** A section with a right-pointing arrow.
- Fault injection - optional:** A section with a right-pointing arrow.
- Container - 1:** A section with a dropdown set to "Essential container" and a "Remove" button.

Implementação de site com AWS ECS

Amazon Elastic Container Service

Express Mode

Clusters

Namespaces

Task definitions

Daemon task definitions [New](#)

Account settings

[AWS Batch](#)

[Amazon ECR](#)

[Repositories](#)

[Online learning workshop](#)

[Documentation](#)

[Discover products](#)

[Subscriptions](#)

▼ Container - 1 [Info](#)

Essential container

[Remove](#)

Container details

Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name

culturadevops-poc-01

Essential container

Yes

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Image URI

repository-uri/image:tag

[Browse ECR images](#)

Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores, colons, periods, forward slashes, and number signs are allowed.

Private registry [Info](#)

Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

Private registry authentication

Port mappings [Info](#)

Add port mappings to allow the container to access ports on the host to send or receive traffic. For port name, a default will be assigned if left blank.

Container port

80

Protocol

TCP

Port name

container-port-protocol

App protocol

HTTP

[Remove](#)

[Add port mapping](#)

Read only root file system [Info](#)

Implementação de site com AWS ECS

Select Amazon ECR image ✕

Private repository

Select the repository containing the image you want to use.

791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01 ✕ ©

Images (1/1)

Find images

< 1 >



Image tag ▾

Image digest [↗](#) ▾

Pushed at ▾

latest

 [sha256:784d159bcaa446c571a382...](#)

April 14, 2026, 15:57 (UTC-3:00)

Select image by

Image digest

Use the SHA256 digest to reference this image.

Image tag

Use a human-readable tag to reference this image.

Cancel

Select image

Implementação de site com AWS ECS

Amazon Elastic Container Service > Create new task definition

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions [New](#)
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Container - 1 [Info](#) Essential container Remove

Container details
Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name Essential container

Yes

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Image URI

Browse ECR images

Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores, colons, periods, forward slashes, and number signs are allowed.

Private registry [Info](#)
Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

Private registry authentication

Port mappings [Info](#)
Add port mappings to allow the container to access ports on the host to send or receive traffic. For port name, a default will be assigned if left blank.

Container port	Protocol	Port name	App protocol	
<input type="text" value="80"/>	<input type="text" value="TCP"/>	<input type="text" value="container-port-protocol"/>	<input type="text" value="HTTP"/>	Remove

Add port mapping

Read only root file system [Info](#)

Implementação de site com AWS ECS

Amazon Elastic Container Service > Create new task definition

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions [New](#)
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Introducing S3 Files support
Amazon S3 Files is a high-performance shared file system that allows you to access S3 data from any AWS compute service. Access your S3 data as a file system in ECS tasks. [Learn more](#) [↗](#)

[Add volume](#)

Volumes from | [Info](#)
Mount data volumes from another container.

[Add volume from](#)

► **Monitoring - optional**
Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.

► **Tags - optional** [Info](#)
Tags help you to identify and organize your task definitions.

[Cancel](#) [Create](#)

Implementação de site com AWS ECS

Amazon Elastic Container Service > Task definitions > culturadevops-poc-01 > Revision 1 > Containers

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions [New](#)
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Task definition successfully created
culturadevops-poc-01:1 has been successfully created. You can use this task definition to deploy a service or run a task.

[View task definition](#) ✕

culturadevops-poc-01:1 Last updated April 14, 2026, 16:44 (UTC-3:00) [Refresh](#) [Deploy](#) [Actions](#) [Create new revision](#)

Overview [Info](#)

ARN arn:aws:ecs:us-east-2:791659331772:task-definition/culturadevops-poc-01:1	Status ACTIVE	Time created April 14, 2026, 16:44 (UTC-3:00)	App environment Fargate
Task role -	Task execution role ecsTaskExecutionRole	Operating system/Architecture Linux/X86_64	Network mode awsvpc
Fault injection Turned off			

[Containers](#) | [JSON](#) | [Task placement](#) | [Volumes \(0\)](#) | [Requires attributes](#) | [Tags](#)

Task size

Task CPU 512 units (0,5 vCPU)	Task memory 1.024 MiB (1 GiB)
---	---

Implementação de site com AWS ECS

The screenshot shows the AWS Management Console interface for the Amazon Elastic Container Service (ECS) Clusters page. The left-hand navigation pane includes links for Express Mode, Clusters (selected), Namespaces, Task definitions, Daemon task definitions (marked as New), Account settings, AWS Batch, Amazon ECR, Repositories, Online learning workshop, Documentation, Discover products, and Subscriptions. The main content area features a notification banner about managed daemons, a search bar containing 'poc' with a 'No matches' result, and a table header for cluster details. A 'Create cluster' button is visible in the top right.

Amazon Elastic Container Service > Clusters

Amazon Elastic Container Service

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Daemon task definitions **New**
- Account settings

- AWS Batch
- Amazon ECR
- Repositories

- Online learning workshop
- Documentation
- Discover products
- Subscriptions

Introducing managed daemons
You can now manage software agents as standalone daemons across container instances in a capacity provider, independent of your applications. [Learn more](#)

Create daemon task definition

Clusters (2) [Info](#)

Last updated April 14, 2026, 16:44 (UTC-3:00) [Refresh](#) **Create cluster**

Search: poc No matches

By default, we only load up to 1.000 clusters at a time.

< 1 > [Settings](#)

Cluster	Services	Tasks	Container instances	CloudWatch monitoring	Capacity provider strateg
---------	----------	-------	---------------------	-----------------------	---------------------------

No matches
We can't find a match.
[Clear filter](#)

Implementação de site com AWS ECS

The screenshot shows the 'Create cluster' page in the AWS Management Console. The left sidebar contains navigation links for Amazon Elastic Container Service, including Express Mode, Clusters, Namespaces, Task definitions, Daemon task definitions, Account settings, AWS Batch, Amazon ECR, Repositories, Online learning workshop, Documentation, Discover products, and Subscriptions. The main content area is titled 'Create cluster' and includes a description of an ECS cluster. It features a 'Cluster configuration' section with a text input for the cluster name 'culturadevops-poc-01' and a note about valid characters. Below this is a section for 'Service Connect defaults - optional'. The 'Infrastructure - advanced' section is expanded, showing options for 'Select a method of obtaining compute capacity'. Three options are listed: 'Fargate only' (selected), 'Fargate and Managed Instances', and 'Fargate and Self-managed instances'.

Amazon Elastic Container Service > Create cluster

Create cluster [Info](#)

An Amazon ECS cluster groups together tasks, and services, and allows for shared capacity and common configurations. All of your tasks, services, and capacity must belong to a cluster.

Cluster configuration

Cluster name

Cluster name must be 1 to 255 characters. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

▶ **Service Connect defaults - optional**

▼ Infrastructure - *advanced* [Info](#)

Configure the manner of obtaining compute resources that will be used to host your application.

Select a method of obtaining compute capacity

Your cluster is automatically configured for AWS Fargate (serverless), but you may choose to add Amazon EC2 instances (servers).

- Fargate only**
Serverless - you don't think about creating or managing servers. Great for most common workloads.
- Fargate and Managed Instances**
Managed instances - Amazon ECS will manage patching and scaling on your behalf while giving you configurability about the types of instances. Great for more advanced workloads.
- Fargate and Self-managed instances**
Self-managed instances - you must ensure the instances are patched and scaled properly, and you have full control over the instances.

Implementação de site com AWS ECS

Amazon Elastic Container Service > Create cluster

Configure the manner of obtaining compute resources that will be used to host your application.

Select a method of obtaining compute capacity
Your cluster is automatically configured for AWS Fargate (serverless), but you may choose to add Amazon EC2 instances (servers).

Fargate only
Serverless - you don't think about creating or managing servers. Great for most common workloads.

Fargate and Managed Instances
Managed instances - Amazon ECS will manage patching and scaling on your behalf while giving you configurability about the types of instances. Great for more advanced workloads.

Fargate and Self-managed instances
Self-managed instances - you must ensure the instances are patched and scaled properly, and you have full control over the instances.

► **Monitoring - optional**
Configure observability, encryption, and logging options to maintain compliance and operational visibility of your container environment.

► **Encryption - optional**
Choose the KMS keys used by tasks running in this cluster to encrypt your storage.

► **Tags - optional** [Info](#)
Tags help you to identify and organize your clusters.

[Cancel](#) [Create](#)

Amazon Elastic Container Service

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- [AWS Batch](#)
- [Amazon ECR](#)
- [Repositories](#)

- [Online learning workshop](#)
- [Documentation](#)
- [Discover products](#)
- [Subscriptions](#)

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters

Cluster culturadevops-poc-01 has been created successfully. [View cluster](#)

Introducing managed daemons
You can now manage software agents as standalone daemons across container instances in a capacity provider, independent of your applications. [Learn more](#) [Create daemon task definition](#)

Clusters (3) [Info](#) Last updated April 14, 2026, 16:46 (UTC-3:00) [Refresh](#) [Create cluster](#)

Q pod X 1 match

By default, we only load up to 1.000 clusters at a time.

Cluster	Services	Tasks	Container instances	CloudWatch monitoring	Capacity provider strategy
culturadevops-poc-01	0	No tasks running	0 EC2	Default	No default found

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions
- Daemon task definitions **New**
- Account settings

AWS Batch [↗](#)

Amazon ECR [↗](#)

Repositories [↗](#)

Online learning workshop [↗](#)

Documentation [↗](#)

Discover products [↗](#)

Subscriptions [↗](#)

culturadevops-poc-01

Last updated April 14, 2026, 16:48 (UTC-3:00) **Actions** **Create with Express Mode**

Cluster overview

ARN am:aws:ecs:us-east-2:791659331772:cluster/culturadevops-poc-01	Status Active	CloudWatch monitoring Default	Registered container instances -
--	--------------------------------	---	--

Services	Tasks
Draining -	Pending -
Active -	Running -

Services | Daemons **New** | Tasks | Infrastructure | Metrics | Scheduled tasks | Configuration | Event history | Tags

Services (0) [Info](#)

Last updated April 14, 2026, 16:48 (UTC-3:00) **Manage tags** **Update** **Delete service** **Create**

Filter launch type Any launch type **Filter scheduling strategy** Any scheduling strategy **Filter resource management type** Any resource management type

<input type="checkbox"/>	Service name	ARN	Status	Schedu...	Laun...	Task de...	Deployments and tasks
--------------------------	--------------	-----	--------	-----------	---------	------------	-----------------------

No services

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Create service



Amazon Elastic Container Service

Express Mode

Clusters

Namespaces

Task definitions

Daemon task definitions [New](#)

Account settings

[AWS Batch](#)

[Amazon ECR](#)

[Repositories](#)

[Online learning workshop](#)

[Documentation](#)

[Discover products](#)

[Subscriptions](#)

Create service [Info](#)

Service details

Task definition family

Select an existing task definition family. To create a new task definition, go to [Task definitions](#).

culturadevops-poc-01

Filter task definitions by family name prefix

culturadevops-poc-01

ecs-cloud-jenkins-agent

futura-td-monolith-dev

jenkins-task-definition-dev

End of all task definition families

Up to 255 letters (uppercase and lowercase), numbers, underscores, and hyphens are allowed. Service names must be unique within a cluster.

Environment

AWS Fargate

Existing cluster

culturadevops-poc-01

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Create service



Amazon Elastic Container Service

Express Mode

Clusters

Namespaces

Task definitions

Daemon task definitions **New**

Account settings

AWS Batch [↗](#)

Amazon ECR [↗](#)

Repositories [↗](#)

Online learning workshop [↗](#)

Documentation [↗](#)

Discover products [↗](#)

Subscriptions [↗](#)

Create service Info

Service details

Task definition family

Select an existing task definition family. To create a new task definition, go to [Task definitions](#).

culturadevops-poc-01

Task definition revision **Latest**

Select the task definition revision from the 100 most recent entries, or enter a revision. Leave the field blank to use the latest revision.

1

Service name

Assign a service name that is unique for this cluster.

culturadevops-poc-01

Up to 255 letters (uppercase and lowercase), numbers, underscores, and hyphens are allowed. Service names must be unique within a cluster.

Environment

AWS Fargate

Existing cluster

culturadevops-poc-01

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Create service

Amazon Elastic Container Service

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Daemon task definitions **New**
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Compute options | [Info](#)

To ensure task distribution across your compute types, use appropriate compute options.

Capacity provider strategy
Specify a launch strategy to distribute your tasks across one or more capacity providers.

Launch type
Launch tasks directly without the use of a capacity provider strategy.

Capacity provider strategy | [Info](#)

Select either your cluster default capacity provider strategy or select the custom option to configure a different strategy.

Use cluster default
No default capacity provider strategy configured for this cluster.

Use custom (Advanced)

Capacity provider | **Base** | [Info](#) | **Weight** | [Info](#)

FARGATE | 0 | 1

[Add capacity provider](#)

You can add up to 1 more capacity provider strategy item.

Platform version | [Info](#)

Specify the platform version on which to run your service.

LATEST

▶ **Troubleshooting configuration - recommended**

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Create service

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions
- Daemon task definitions **New**
- Account settings

AWS Batch [↗](#)

Amazon ECR [↗](#)

Repositories [↗](#)

Online learning workshop [↗](#)

Documentation [↗](#)

Discover products [↗](#)

Subscriptions [↗](#)

Deployment configuration

Scheduling strategy | [Info](#)

Replica
Place and maintain a desired number of tasks across your cluster.

Daemon
Place and maintain one copy of your task on each container instance.

Desired tasks
Specify the number of tasks to launch.

Availability Zone rebalancing | [Info](#)

Turn on Availability Zone rebalancing
Amazon ECS automatically detects Availability Zone imbalances in task distributions across an ECS service, and evenly redistributes ECS service tasks across Availability Zones.

Health check grace period | [Info](#)

seconds

► **Deployment options**

► **Deployment failure detection** [Info](#)

Implementação de site com AWS ECS

VPC [Info](#)

Select a VPC to use for your Amazon ECS resources.

vpc-0c1801daaecdb9c21
futura-vpc-app-dev



Create a new VPC [↗](#)

Subnets

Choose the subnets within the VPC that the task scheduler should consider for placement.

Choose subnets



Clear current selection

subnet-0a3a0a4ea17a17ec5 ✕
futura-pub-subnet-app-1-dev
us-east-2a 11.0.0.64/27

Security group [Info](#)

Choose an existing security group or create a new security group.

- Use an existing security group
- Create a new security group

Security group name

Choose an existing security group.

Choose security groups



sg-0e491e6a1c4427834 ✕
futura-sg-ecs-service-v2-dev | futura-sg-ecs-service-v2-dev

Public IP [Info](#)

Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).

- Turned on

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Create service

Amazon Elastic Container Service

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- [AWS Batch](#)
- [Amazon ECR](#)
- [Repositories](#)

- [Online learning workshop](#)
- [Documentation](#)
- [Discover products](#)
- [Subscriptions](#)

Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

▶ **VPC Lattice - optional** [Info](#)
Fully managed application networking service to connect, secure, and monitor your services across multiple accounts and virtual private clouds (VPCs). When you use VPC Lattice, there is a cost associated with it.

▶ **Service auto scaling - optional**
Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

▶ **Volume - optional** [Info](#)
Configure a data volume to provide additional storage for the containers in the task.

▶ **Tags - optional** [Info](#)
Tags help you to identify and organize your resources.

[Cancel](#) [Create](#)

Implementação de site com AWS ECS

The screenshot displays the AWS ECS console interface. The breadcrumb navigation at the top reads: Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services. The main header for the cluster 'culturadevops-poc-01' shows it was last updated on April 14, 2026, at 16:58 (UTC-3:00). There are 'Actions' and 'Create with Express Mode' buttons.

Cluster overview

ARN arn:aws:ecs:us-east-2:791659331772:cluster/culturadevops-poc-01	Status Active	CloudWatch monitoring Default	Registered container instances -
---	-------------------------	---	--

Services

Draining -	Active -	Pending -	Running -
----------------------	--------------------	---------------------	---------------------

Tasks

Pending -	Running -
---------------------	---------------------

The console also features a navigation bar with tabs for Services, Daemons, Tasks, Infrastructure, Metrics, Scheduled tasks, Configuration, Event history, and Tags. Below this, the 'Services (1) Info' section shows a table with filters and a list of services.

Services (1) Info

Filters: Filter services by value, Filter launch type: Any launch type, Filter scheduling strategy: Any scheduling strategy, Filter resource management type: Any resource management type.

Service name	ARN	Status	Schedu...	Laun...	Task de...	Deployments and tasks
culturadevops-poc-01	arn:aws:ecs:us-e	Active	REPLICA	-	culturade...	0/1 Tasks running

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services

Amazon Elastic Container Service

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- [AWS Batch](#)
- [Amazon ECR](#)
- [Repositories](#)

- [Online learning workshop](#)
- [Documentation](#)
- [Discover products](#)
- [Subscriptions](#)

culturadevops-poc-01

Last updated April 14, 2026, 16:59 (UTC-3:00) [Actions](#) [Create with Express Mode](#)

Cluster overview

ARN arn:aws:ecs:us-east-2:791659331772:cluster/culturadevops-poc-01	Status Active	CloudWatch monitoring Default	Registered container instances -
---	-------------------------	---	--

Services

Draining -	Active 1	Pending -	Running 1
----------------------	--------------------	---------------------	---------------------

Tasks

Pending -	Running 1
---------------------	---------------------

[Services](#) | [Daemons](#) [New](#) | [Tasks](#) | [Infrastructure](#) | [Metrics](#) | [Scheduled tasks](#) | [Configuration](#) | [Event history](#) | [Tags](#)

Services (1) Info

Last updated April 14, 2026, 16:59 (UTC-3:00) [Manage tags](#) [Update](#) [Delete service](#) [Create](#)

Filter services by value Filter launch type Any launch type Filter scheduling strategy Any scheduling strategy Filter resource management type Any resource management type

<input type="checkbox"/>	Service name	ARN	Status	Schedu...	Laun...	Task de...	Deployments and tasks
<input type="checkbox"/>	culturadevops-poc-01	arn:aws:ecs:us-e	Active	REPLICA	-	culturade...	1/1 Tasks running

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Tasks

Cluster overview

ARN arn:aws:ecs:us-east-2:791659331772:cluster/culturadevops-poc-01	Status Active	CloudWatch monitoring Default	Registered container instances -
Services	Tasks		
Draining -	Active 1	Pending -	Running 1

Services | Daemons **New** | **Tasks** | Infrastructure | Metrics | Scheduled tasks | Configuration | Event history | Tags

Tasks (1)

Last updated April 14, 2026, 17:26 (UTC-3:00) Refresh Manage tags Stop Run new task

Filter tasks by property or value Filter desired status: Running Filter launch type: Any launch type < 1 > Settings

<input type="checkbox"/>	Task	Last status	Desired status	Task definition	Health status
<input type="checkbox"/>	5cb9f25989a3417fa4062b505be1de7f	Running	Running	culturadevops-poc-01:1	Unknown

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Tasks > 5cb9f25989a3417fa4062b505be1de7f > Configuration

Configuration Metrics Logs Networking Volumes (0) Tags

Task overview

ARN arn:aws:ecs:us-east-2:791659331772:task/culturadevops-poc-01/5cb9f25989a3417fa4062b505be1de7f	Last status Running	Desired status Running	Started/Created at April 14, 2026, 17:25 (UTC-3:00) April 14, 2026, 17:24 (UTC-3:00)
---	-------------------------------	----------------------------------	---

Started by
ecs-svc/7662231768067865890

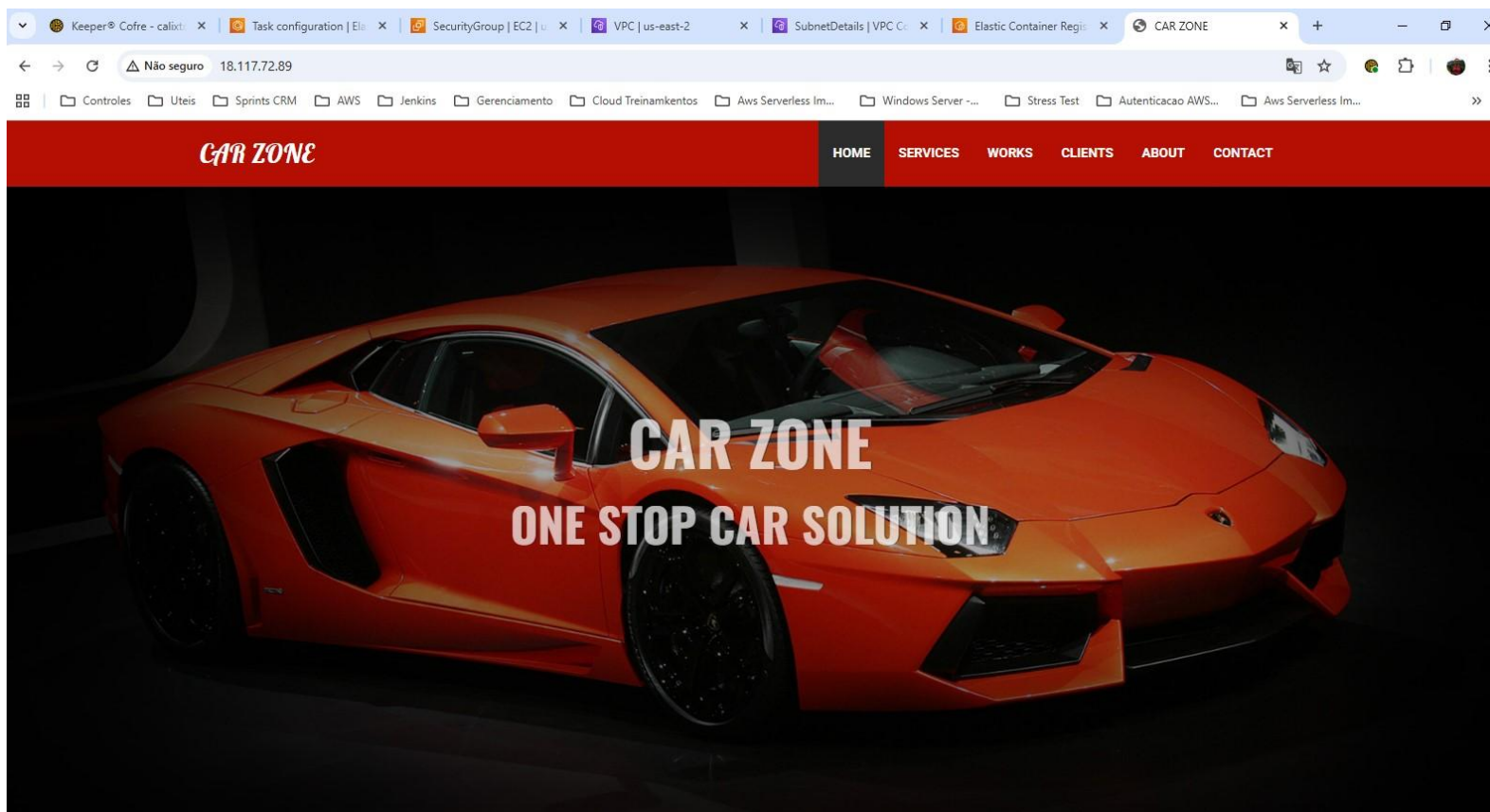
Container details for culturadevops-poc-01

Details Log configuration Restart policy **Network bindings** Docker labels and hosts Environment variables and files Volume c >

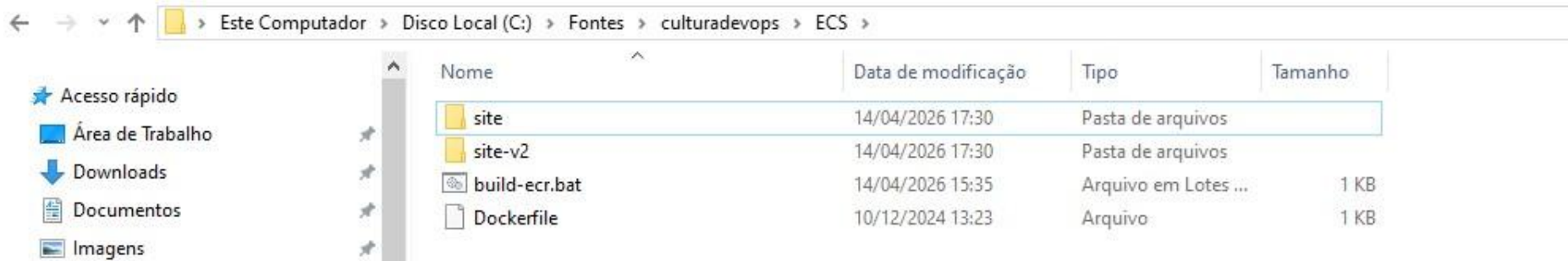
Network bindings

Host port	Container port	Protocol	External link
80	80	tcp	18.117.72.89:80 open address

Implementação de site com AWS ECS



Implementação de site com AWS ECS



The screenshot shows a Windows File Explorer window with the address bar displaying the path: `> Este Computador > Disco Local (C:) > Fontes > culturadevops > ECS >`. The left sidebar shows quick access links for `Área de Trabalho`, `Downloads`, `Documentos`, and `Imagens`. The main pane displays a table of files and folders:

Nome	Data de modificação	Tipo	Tamanho
site	14/04/2026 17:30	Pasta de arquivos	
site-v2	14/04/2026 17:30	Pasta de arquivos	
build-ecr.bat	14/04/2026 15:35	Arquivo em Lotes ...	1 KB
Dockerfile	10/12/2024 13:23	Arquivo	1 KB

Implementação de site com AWS ECS



Dockerfile - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
from nginx
```

```
copy ./site-v2 /usr/share/nginx/html
```

```
Expose 80
```

```
ECS .\build-ecr
C:\Fontes\culturadevops\ECS>aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 791659331772.dkr.ecr.us-east-2.amazonaws.com
Login Succeeded

C:\Fontes\culturadevops\ECS>docker build -t culturadevops-poc-01 .
[+] Building 2.1s (8/8) FINISHED
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 96B 0.0s
=> [internal] load metadata for docker.io/library/nginx:latest 1.7s
=> [auth] library/nginx:pull token for registry-1.docker.io 0.0s
=> [internal] load build context 0.9s
=> => transferring context: 2.68MB 0.3s
=> CACHED [1/2] FROM docker.io/library/nginx@sha256:7f6adca1fc6c29c8dc49a2e90837a18ba20dc266baaed0988e9f64d8d8b85ba0 0.0s
=> [2/2] COPY ./site-v2 /usr/share/nginx/html 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> writing image sha256:9b123cf6c9a15c83f863da66477cd393405174d313b92d471b49ff48a99f977c 0.0s
=> naming to docker.io/library/culturadevops-poc-01 0.0s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

C:\Fontes\culturadevops\ECS>docker tag culturadevops-poc-01:latest 791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01:latest

C:\Fontes\culturadevops\ECS>docker push 791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01:latest
The push refers to repository [791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01]
0a46ed6c4242: Pushed
9d099dfdcffeb: Layer already exists
5ca2f881e59f: Layer already exists
4b018d686a3c: Layer already exists
3fd3ceafa86d: Layer already exists
4ef7ac306afb: Layer already exists
a4b536aa4bf8: Layer already exists
60e78ddddd9ea: Layer already exists
latest: digest: sha256:056520b584b56f23e577fb3cd8054aba779c4a47ae168dc401d468b2aa5da7e9 size: 1989
Rogerio P. Calixto ECS
```

in push at 17:33:00

Implementação de site com AWS ECS

Amazon ECR > Private registry > Repositories > culturadevops-poc-01

Amazon Elastic Container Service

Private registry

- Repositories
 - culturadevops-poc-01**
- Features & Settings

Public registry

- Repositories
- Settings

ECR public gallery [↗](#)

Amazon ECS [↗](#)

Amazon EKS [↗](#)

Getting started [↗](#)

Documentation [↗](#)

culturadevops-poc-01

Summary | **Images** | Lifecycle policy | Permissions | Repository tags

Images (2) [Info](#) [Refresh](#) [Delete](#) [Copy URI](#) [Details](#) [Scan](#) [View push commands](#)

Filter active images

<input type="checkbox"/>	Image tags	Type	Created at	Image size (MB)	Image digest	Last pulled at
<input type="checkbox"/>	latest	Image	April 14, 2026, 17:32:59 (UTC-03)	65.53	sha256:056520b5...	-
<input type="checkbox"/>	-	Image	April 14, 2026, 15:57:42 (UTC-03)	66.65	sha256:784d159b...	April 14, 2026, 16:59:14 (UTC-03)

Implementação de site com AWS ECS

Amazon Elastic Container Service > Task definitions > culturadevops-poc-01

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions New
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)

Service updated: culturadevops-poc-01:culturadevops-poc-01

culturadevops-poc-01 (1/1) Info

Last updated April 14, 2026, 17:34 (UTC-3:00) [↻](#) [Deploy](#) [Actions](#) [Create new revision](#)

Filter task definition revisions by value

Filter status: Active

<input checked="" type="checkbox"/>	Task definition: revision	Status
<input checked="" type="checkbox"/>	culturadevops-poc-01:1	Active

Implementação de site com AWS ECS

Amazon Elastic Container Service > Task definitions > culturadevops-poc-01 > Revision 1 > Create revision

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions [New](#)
- Account settings

- [AWS Batch](#)
- [Amazon ECR](#)
- [Repositories](#)

- [Online learning workshop](#)
- [Documentation](#)
- [Discover products](#)
- [Subscriptions](#)

Task execution role [Info](#)

A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't already have a task execution IAM role created, we can create one for you.

ecsTaskExecutionRole [↻](#) [Create new role](#)

▶ **Task placement - optional**

▶ **Fault injection - optional**

▼ **Container - 1** [Info](#) Essential container [Remove](#)

Container details

Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name Essential container

culturadevops-poc-01 Yes

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Image URI

791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01@sha256:07e2760c00168e34a55c65c7857fec15e [Browse ECR images](#)

Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores, colons, periods, forward slashes, and number signs are allowed.

Private registry [Info](#)

Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

Private registry authentication

Implementação de site com AWS ECS

Select Amazon ECR image



Private repository

Select the repository containing the image you want to use.

791659331772.dkr.ecr.us-east-2.amazonaws.com/culturadevops-poc-01



Images (1/3)

Find images

< 1 >



Image tag



Image digest



Pushed at



latest

sha256:07e2760c00168e34a55c65...

April 14, 2026, 17:46 (UTC-3:00)

-

sha256:056520b584b56f23e577fb...

April 14, 2026, 17:32 (UTC-3:00)

-

sha256:784d159bcaa446c571a382...

April 14, 2026, 15:57 (UTC-3:00)

Select image by

Image digest

Use the SHA256 digest to reference this image.

Image tag

Use a human-readable tag to reference this image.

Cancel

Select image

Implementação de site com AWS ECS

Amazon Elastic Container Service > Task definitions > culturadevops-poc-01 > Revision 1 > Create revision

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions New
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Introducing S3 Files support
Amazon S3 Files is a high-performance shared file system that allows you to access S3 data from any AWS compute service. Access your S3 data as a file system in ECS tasks. [Learn more ↗](#)

[Add volume](#)

Volumes from [Info](#)
Mount data volumes from another container.

[Add volume from](#)

► **Monitoring - optional**
Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.

► **Tags - optional** [Info](#)
Tags help you to identify and organize your task definitions.

[Cancel](#) [Create](#)

Implementação de site com AWS ECS

Amazon Elastic Container Service > Task definitions > culturadevops-poc-01 > Revision 2 > Containers

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions**
- Daemon task definitions [New](#)
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Task definition successfully created
culturadevops-poc-01:2 has been successfully created. You can use this task definition to deploy a service or run a task. [View task definition](#)

Notifications 0 0 2 0 0

culturadevops-poc-01:2

Last updated April 14, 2026, 17:36 (UTC-3:00) [Refresh](#) [Deploy](#) [Actions](#) [Create new revision](#)

Overview [Info](#)

ARN arn:aws:ecs:us-east-2:791659331772:task-definition/culturadevops-poc-01:2	Status ACTIVE	Time created April 14, 2026, 17:36 (UTC-3:00)	App environment Fargate
Task role -	Task execution role ecsTaskExecutionRole	Operating system/Architecture Linux/X86_64	Network mode awsvpc
Fault injection Turned off			

Containers | JSON | Task placement | Volumes (0) | Requires attributes | Tags

Task size

Task CPU	Task memory
----------	-------------

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- [AWS Batch](#)
- [Amazon ECR](#)
- [Repositories](#)

- [Online learning workshop](#)
- [Documentation](#)
- [Discover products](#)
- [Subscriptions](#)

Cluster overview

ARN arn:aws:ecs:us-east-2:791659331772:cluster/culturadevops-poc-01	Status Active	CloudWatch monitoring Default	Registered container instances -
Services Draining -	Active 1	Tasks Pending -	Running 1

[Services](#) | [Daemons](#) **New** | [Tasks](#) | [Infrastructure](#) | [Metrics](#) | [Scheduled tasks](#) | [Configuration](#) | [Event history](#) | [Tags](#)

Services (1/1) [Info](#)

Last updated April 14, 2026, 17:36 (UTC-3:00) [Manage tags](#) [Update](#) [Delete service](#) [Create](#)

Filter services by value Filter launch type Any launch type Filter scheduling strategy Any scheduling strategy Filter resource management type Any resource management type

<input checked="" type="checkbox"/>	Service name	ARN	Status	Schedu...	Laun...	Task definition	Deployments and tasks
<input checked="" type="checkbox"/>	culturadevops-poc-01	arn:aws:ecs:us-e	Active	REPLICA	-	culturadevops-poc-01:1	1/1 Tasks runnin

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services > culturadevops-poc-01 > Update

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Task definition successfully created
culturadevops-poc-01:2 has been successfully created. You can use this task definition to deploy a service or run a task. [View task definition](#)

Notifications 0 0 2 0 0

Update culturadevops-poc-01 [Info](#)

Deployment configuration

Force new deployment

Task definition family
Select an existing task definition family. To create a new task definition, go to [Task definitions](#).

culturadevops-poc-01 [↻](#)

Task definition revision
Select the task definition revision from the 100 most recent entries, or enter a revision. Leave the field blank to use the latest revision.

1 [✕](#) [↻](#)

Use: 1
2 (LATEST)
1

Specify the number of tasks to launch.

1

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services > culturadevops-poc-01 > Update

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Task definition successfully created
culturadevops-poc-01:2 has been successfully created. You can use this task definition to deploy a service or run a task. [View task definition](#)

Notifications 0 0 2 0 0 0

Update culturadevops-poc-01 [Info](#)

Deployment configuration

Force new deployment

Task definition family
Select an existing task definition family. To create a new task definition, go to [Task definitions](#).

culturadevops-poc-01

Task definition revision [Latest](#)
Select the task definition revision from the 100 most recent entries, or enter a revision. Leave the field blank to use the latest revision.

2

Scheduling strategy
REPLICA

Desired tasks
Specify the number of tasks to launch.

1

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services > culturadevops-poc-01 > Update

Amazon Elastic Container Service

- Express Mode
- Clusters**
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- [AWS Batch](#)
- [Amazon ECR](#)
- [Repositories](#)

- [Online learning workshop](#)
- [Documentation](#)
- [Discover products](#)
- [Subscriptions](#)

Fully managed application networking service to connect, secure, and monitor your services across multiple accounts and virtual private clouds (VPCs). When you use VPC Lattice, there is a cost associated with it.

► **Load balancing - optional**
Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

► **Service auto scaling - optional**
Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

► **Volume - optional** [Info](#)
Configure a data volume to provide additional storage for the containers in the task.

► **Tags - optional**
Tags help you to identify and organize your clusters.

[Cancel](#) [Update](#)

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- [AWS Batch](#)
- [Amazon ECR](#)
- [Repositories](#)

- [Online learning workshop](#)
- [Documentation](#)
- [Discover products](#)
- [Subscriptions](#)

Cluster overview

ARN <code>arn:aws:ecs:us-east-2:791659331772:cluster/culturadevops-poc-01</code>	Status Active	CloudWatch monitoring Default	Registered container instances -
---	-------------------------	----------------------------------	-------------------------------------

Services

Draining -	Active 1	Pending -	Running 1
---------------	-------------	--------------	--------------

Services (1) Info

Last updated April 14, 2026, 17:39 (UTC-3:00) [Manage tags](#) [Update](#) [Delete service](#) [Create](#)

[Filter launch type](#) Any launch type [Filter scheduling strategy](#) Any scheduling strategy [Filter resource management type](#) Any resource management type < 1 >

<input type="checkbox"/>	Service name	ARN	Status	Schedu...	Laun...	Task definition	Deployments and tasks
<input type="checkbox"/>	culturadevops-poc-01	<code>arn:aws:ecs:us-e</code>	Active	REPLICA	-	culturadevops-poc-01:2	1/1 Tasks running

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Tasks

Amazon Elastic Container Service

Express Mode
Clusters
Namespaces
Task definitions
Daemon task definitions [New](#)
Account settings

AWS Batch [↗](#)
Amazon ECR [↗](#)
Repositories [↗](#)

Draining - Active 1 Pending - Running 1

Services | Daemons [New](#) | **Tasks** | Infrastructure | Metrics | Scheduled tasks | Configuration | Event history | Tags

Tasks (4) Last updated April 14, 2026, 17:41 (UTC-3:00) [Refresh](#) [Manage tags](#) [Stop](#) [Run new task](#)

Filter desired status Any desired status **Filter launch type** Any launch type < 1 > ⚙️

<input type="checkbox"/>	Task	Last status	Desired status	Task definition	Health status
<input type="checkbox"/>	42e480caf73445bbbd33633f17c2111a	Running	Running	culturadevops-poc-01:2	Unknown
<input type="checkbox"/>	5cb9f25989a3417fa4062b505be1de7f	Stopped ...	Stopped	culturadevops-poc-01:1	Unknown

Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Tasks > 42e480caf73445bbbd33633f17c2111a > Configuration

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions
- Daemon task definitions **New**
- Account settings

- AWS Batch ↗
- Amazon ECR ↗
- Repositories ↗

- Online learning workshop ↗
- Documentation ↗
- Discover products ↗
- Subscriptions ↗

Service updated: culturadevops-poc-01:culturadevops-poc-01

Notifications 0 0 2 0 0

42e480caf73445bbbd33633f17c2111a

Last updated April 14, 2026, 17:42 (UTC-3:00) [Refresh](#) [Stop](#)

Configuration | Metrics | Logs | Networking | Volumes (0) | Tags

Task overview

Container details for culturadevops-poc-01

Details | Log configuration | Restart policy | **Network bindings** | Docker labels and hosts | Environment variables and files | Volume c >

Network bindings

Host port	Container port	Protocol	External link
80	80	tcp	18.116.19.71:80 open address ↗

Implementação de site com AWS ECS



Implementação de site com AWS ECS

Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Services > culturadevops-poc-01 > Update

Amazon Elastic Container Service

- Express Mode
- Clusters
- Namespaces
- Task definitions
- Daemon task definitions [New](#)
- Account settings

- AWS Batch [↗](#)
- Amazon ECR [↗](#)
- Repositories [↗](#)

- Online learning workshop [↗](#)
- Documentation [↗](#)
- Discover products [↗](#)
- Subscriptions [↗](#)

Deployment configuration

Force new deployment

Task definition family
Select an existing task definition family. To create a new task definition, go to [Task definitions](#) [↗](#).

culturadevops-poc-01 [↻](#)

Task definition revision [Latest](#)
Select the task definition revision from the 100 most recent entries, or enter a revision. Leave the field blank to use the latest revision.

Q 4 [✕](#) [↻](#)

Scheduling strategy
REPLICA

Desired tasks
Specify the number of tasks to launch.

0 [↕](#)

Availability Zone rebalancing | [Info](#)
 Turn on Availability Zone rebalancing
Amazon ECS automatically detects Availability Zone imbalances in task distributions across an ECS service, and evenly redistributes ECS service tasks across Availability Zones.

Health check grace period | [Info](#)

Implementação de site com AWS ECS

The screenshot shows the AWS Management Console interface for the Amazon Elastic Container Service (ECS) console. The breadcrumb navigation indicates the path: Amazon Elastic Container Service > Clusters > culturadevops-poc-01 > Tasks.

Cluster overview

- ARN:** am:aws:ecs:us-east-2:791659331772:cluster/culturadevops-poc-01
- Status:** Active
- CloudWatch monitoring:** Default
- Registered container instances:** -
- Services:** Draining: -, Active: 1
- Tasks:** Pending: -, Running: -

Tasks (0)

Last updated April 14, 2026, 18:12 (UTC-3:00)

Filter tasks by property or value

Filter desired status: Running

Filter launch type: Any launch type

Manage tags | Stop | Run new task

Task | Last status | Desired status | Task definition | Health status

No tasks
No tasks to display.
Run new task

Implementação de site com AWS ECS

