

Create New Task Definition

1. Select EC2 launch type



Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE	EC2
	
Price based on task size	Price based on resource usage
Requires network mode awsvpc	Multiple network modes available
AWS-managed infrastructure, no Amazon EC2 instances to manage	Self-managed infrastructure using Amazon EC2 instances

*Required

Cancel

Next step

2. Click “Configure via JSON”

Container N...	Image	Hard/Soft m...	CPU Units	GPU	Essential	
No results						

Constraint

Task placement constraints allow you to filter the container instances used for the placement of your tasks using built-in or custom attributes. The service scheduler first filters the container instances that match the constraints and then applies the placement strategy to place the task.

Type	Expression
Add constraint	

Volumes ?

[Add volume](#)

[Configure via JSON](#)

Tags

Key	Value
<input type="text" value="Add key"/>	<input type="text" value="Add value"/>

*Required

Cancel Previous Create

3. Empty the contents and paste the contents of this [file](#).

4. Edit “Task Definition Name”, “Task Role” and “Task Execution Role” according to your requirements.

Step 2: Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task Definition Name* ⓘ

Requires Compatibilities* EC2

Task Role ⓘ
Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#).

Network Mode ⓘ
If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.

Task execution role ⓘ

- Click “deepfence_agent” container on the “Container Definitions” heading and replace “Secrets Manager ARN” with your deepfence docker hub secret ARN.

Edit container ⓘ

▼ Standard

Container name* ⓘ

Image* ⓘ
Custom image format: [registry-url]/[namespace]/[image]:[tag]

Private repository authentication* ☒ ⓘ

Secrets manager ARN ⓘ
The task role specified in the task definition must contain KMS key decrypt permissions ("kms:Decrypt"). Otherwise, image authentication fails, preventing the container image pull. [Learn more](#)

Memory Limits (MiB)* ⓘ
[Add Hard limit](#)
Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the 'memory' and 'memoryReservation' parameters, respectively, in task definitions.

5. Change the ip address “0.0.0.0” to IP address of Deepfence Management Console in following two places.

Task m

0

Task C

0

C

Const

Task pla

attribute

to place

Type

+ Ac

Volume

Name

Host

Source

Edit container

Entry point

/usr/local/bin/start_services.sh

?

Command

-p 0.0.0.0 -f,appsec,-s,1,-c,1

?

Working directory

/home/deepfence

?

Environment variables

You may also designate AWS Systems Manager Parameter Store keys or ARNs using the 'valueFrom' field. ECS will inject the value into containers at run-time.

Key

DF_BACKEND_IP

Value

0.0.0.0

✕

DF_BACKEND_PORT

Value

8010

✕

DF_CAPTURE_INTF

Value

any

✕

DF_KUBERNETES_ON

Value

N

✕

DF_KUBERNETES_VERSION

Value

Add value

✕

* Required

Cancel

Update

6. Save the task definition.

Deploy Deepfence Agent Service

1. In your cluster, click “Create” button in “Service” tab.
2. Set launch type as EC2, service type as DAEMON, minimum healthy percent as 99

Launch type ☐ FARGATE ☒ EC2 ⓘ

Task Definition Family
deepfence-agent-task ▼ Enter a value

Revision
3 (latest) ▼

Cluster ecs-test ▼ ⓘ

Service name deepfence_agent_ecs ⓘ

Service type* ☐ REPLICA ☒ DAEMON ⓘ

Number of tasks Automatic ⓘ

Minimum healthy percent 99 ⓘ

Maximum percent 100 ⓘ

Deployments

Choose a deployment option for the service.

Deployment type* ☒ Rolling update ⓘ

☐ Blue/green deployment (powered by AWS CodeDeploy) ⓘ

This sets AWS CodeDeploy as the deployment controller for the service. A

3. In the next page, set Load Balancer Type as None.
4. Set everything else to default values and save.