

ECS Tasks Definitions

IAM Role	1
Role 1: netflux-task-role	1
Role 2: ecs-task-execution-role	3
ECS Task Definitions	4
JSON reference	7
CloudWatch Log Group	9

IAM Role

ECS Fargate is the serverless container orchestrator. It needs 2 IAM roles.

1. A role for our netflux application to call **secrets manager** to get the RDS credentials.
2. A role for Fargate cluster to pull the docker image and run the application.

Let's create these one by one.

Role 1: netflux-task-role

- This is for our application

Trusted entity type

- ☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ☐ **Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

- Our ECS Task would be making AWS API calls

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Elastic Container Service ▼

Choose a use case for the specified service.


Use case

- ☐ Elastic Container Service
Allows ECS to create and manage AWS resources on your behalf.
- ☐ Elastic Container Service Autoscale
Allows Auto Scaling to access and update ECS services.
- ☒ Elastic Container Service Task
Allows ECS tasks to call AWS services on your behalf.
- ☐ EC2 Role for Elastic Container Service
Allows EC2 instances in an ECS cluster to access ECS.

- We need these 2 permissions
 - ECS tasks will have to register themselves to Target Groups
 - They will have to access Secrets Manager for credentials

Step 2: Add permissions

Permissions policy summary

Policy name 	Type	Attached as
AmazonEC2ContainerServiceRole	AWS managed	Permissions policy
SecretsManagerReadWrite	AWS managed	Permissions policy

- Let's give a name for the role

Role details

Role name

Enter a meaningful name to identify this role.

netflux-task-role

Maximum 64 characters. Use alphanumeric and '+=,.,@-_' characters.

Description

Role 2: ecs-task-execution-role

- This is for Fargate cluster

Trusted entity type

☒ **AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**
Allows users federated by the web identity provider to assume perform actions in this account.

☐ **SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

- Select ECS & Task Execution Role

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Elastic Container Service

Choose a use case for the specified service.
Use case

☐ Elastic Container Service for Service Connect Fargate
Allows ECS to create and manage AWS service resources required to encrypt Amazon ECS Service Connect

☒ **Task Execution Role for Elastic Container Service**
Allows access to other AWS service resources that are required to run Amazon ECS tasks.

☐ EC2 Role for ECS Managed Instances

- role name

Role details

Role name
Enter a meaningful name to identify this role.


ecs-task-execution-role

Maximum 64 characters. Use alphanumeric and '+=,.,@-_' characters.

- Create the Role. Once created, ensure that you have this policy

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

<input type="checkbox"/>	Policy name ↗	Type
<input type="checkbox"/>	 AmazonECSTaskExecutionRolePolicy	AWS managed

ECS Task Definitions

We need to create task-definitions for every single microservice we have! This task definition is similar to Kubernetes deployment yaml file. It contains

- image name
 - CPU / memory
 - IAM role
- Create task definition for movie-service

Task definition configuration

Task definition family [Info](#)

Specify a unique task definition family name.

movie-service

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

- Select Fargate

Launch type [Info](#)

Selection of the launch type will change task definition parameters.

☒ **AWS Fargate**

Serverless compute for containers.

☐ **Amazon EC2 instances**

Self-managed infrastructure using Amazon EC2 instances.

- CPU architecture

OS, Architecture, Network mode

Network mode is used for tasks and is dependent on the compute type sel

Operating system/Architecture [Info](#)

Linux/X86_64 ▼

- CPU / Memory requirements

Task size [Info](#)

Specify the amount of CPU and memory to reserve for your task.

CPU

.5 vCPU ▼

Memory

1 GB ▼

- IAM Roles

▼ Task roles - *conditional*

Task role [Info](#)

A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role from the [IAM console](#)

netflux-task-role ▼

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 exec

ecs-task-execution-role ▼

- Provide docker image details. Copy Image URI from ECR or Select image tag

Select Amazon ECR image

Private repository

Select the repository containing the image you want to use.

127549748682.dkr.ecr.us-east-1.amazonaws.com/movie-service

Images (1/3)

Find images

	Image tag	Image digest	Pushed a
<input checked="" type="radio"/>	latest	sha256:2648bdeca7ebd7cc8c3f0ab...	Novembe
<input type="radio"/>	-	sha256:b299db7fe63e06c765a31d...	Novembe
<input type="radio"/>	-	sha256:f688f5bb0127f2793f2b61...	Novembe

Select image by

- ☐ Image digest
Use the SHA256 digest to reference this image.
- ☒ Image tag
Use a human-readable tag to reference this image.

Image tag

latest

- container port 8080

Port mappings

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	
8080	TCP	container-port-protocol	HTTP	Remove

- Environment variables

Environment variables - optional

Environment variables

Add individually

Add environment variables using plain text values or secrets from AWS Secrets Manager or Parameter Store.


Key	Value type	Value
SPRING_PROFILES_ACTIVE	Value	prod

Add environment variable

- Everything else is optional. Click on “Create”
- Repeat the same for “customer-service”

- We should have 2 task definitions created

Amazon Elastic Container Service > Task definitions

Task definitions (2) [Info](#)  [Deploy](#) [Create new revision](#)

Filter by status

	Task definition	Status of last revision
<input type="radio"/>	customer-service	✓ ACTIVE
<input type="radio"/>	movie-service	✓ ACTIVE

JSON reference

```
{
  "family": "movie-service",
  "containerDefinitions": [
    {
      "cpu": 0,
      "environment": [
        {
          "name": "SPRING_PROFILES_ACTIVE",
          "value": "prod"
        }
      ],
      "essential": true,
      "image": "127549748682.dkr.ecr.us-east-1.amazonaws.com/movie-service:latest",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/movie-service",
          "awslogs-create-group": "true",
          "awslogs-region": "us-east-1",
          "awslogs-stream-prefix": "ecs"
        }
      },
      "mountPoints": [],
      "name": "movie-service",
      "portMappings": [
        {
          "containerPort": 8080,
          "hostPort": 8080,
          "protocol": "tcp"
        }
      ]
    }
  ]
}
```

```

    }
    ],
    "systemControls": [],
    "volumesFrom": []
  }
],
"taskRoleArn": "arn:aws:iam::127549748682:role/netflux-task-role",
"executionRoleArn": "arn:aws:iam::127549748682:role/ecs-task-execution-role",
"networkMode": "awsvpc",
"volumes": [],
"placementConstraints": [],
"requiresCompatibilities": [
  "FARGATE"
],
"cpu": "512",
"memory": "1024"
}

```

```

{
  "family": "customer-service",
  "containerDefinitions": [
    {
      "cpu": 0,
      "environment": [
        {
          "name": "SPRING_PROFILES_ACTIVE",
          "value": "prod"
        }
      ],
      "essential": true,
      "image":
"127549748682.dkr.ecr.us-east-1.amazonaws.com/customer-service:latest",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/customer-service",
          "awslogs-create-group": "true",
          "awslogs-region": "us-east-1",
          "awslogs-stream-prefix": "ecs"
        }
      },
      "mountPoints": [],
      "name": "customer-service",
      "portMappings": [
        {
          "containerPort": 8080,
          "hostPort": 8080,
          "protocol": "tcp"
        }
      ],
      "systemControls": [],
    }
  ]
}

```



```

        "volumesFrom": []
    },
    "taskRoleArn": "arn:aws:iam::127549748682:role/netflux-task-role",
    "executionRoleArn": "arn:aws:iam::127549748682:role/ecs-task-execution-role",
    "networkMode": "awsvpc",
    "volumes": [],
    "placementConstraints": [],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "512",
    "memory": "1024"
}

```

CloudWatch Log Group

- CloudWatch is the monitoring service provided by AWS. When applications run, they generate logs that need a place to be stored and managed.
- For this purpose, we create Log Groups in CloudWatch corresponding to each ECS task definition, such as:
 - /ecs/movie-service
 - /ecs/customer-service
- These log groups help organize and monitor logs for each service independently.

The screenshot shows the AWS CloudWatch console interface. On the left is a navigation sidebar with 'CloudWatch' at the top, followed by 'Log groups' (selected), 'Dashboards', 'Alarms', and 'Logs'. The main content area is titled 'Log groups (2)' and includes a search bar with the placeholder 'Filter log groups or try pattern search'. Below the search bar is a table listing two log groups:

<input type="checkbox"/>	Log group	Log class	Anomaly d...
<input type="checkbox"/>	/ecs/movie-service	Standard	Configure
<input type="checkbox"/>	/ecs/customer-service	Standard	Configure