Farget Role / Cluster / Tasks Definition

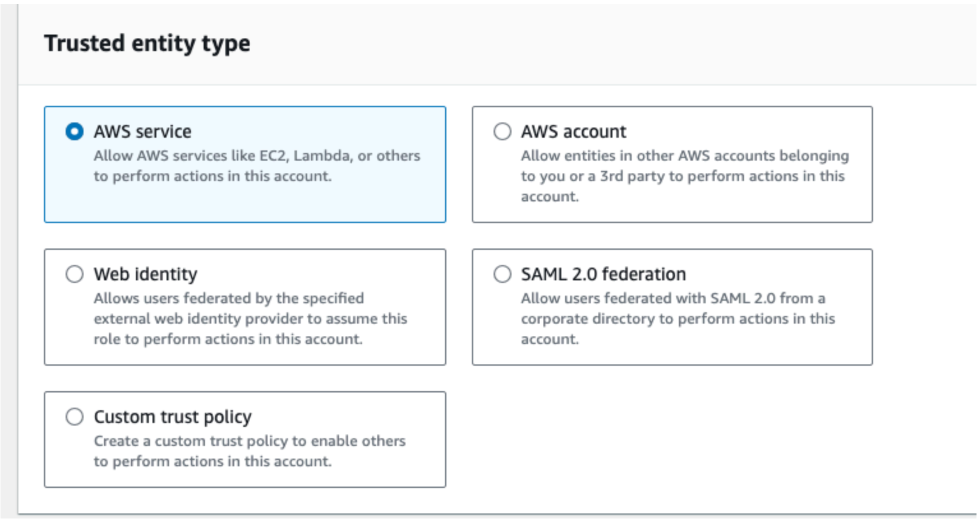
IAM

ECS + Fargate is the serverless container orchestrator.

- It needs an IAM role to pull docker images etc.
- Our app (customer-service/movie-service) needs an IAM role to access secrets manager

We need 2 IAM roles.

- AWS will create 1 role itself.
- Let's create another role. This is for our app,



Trusted entity type

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

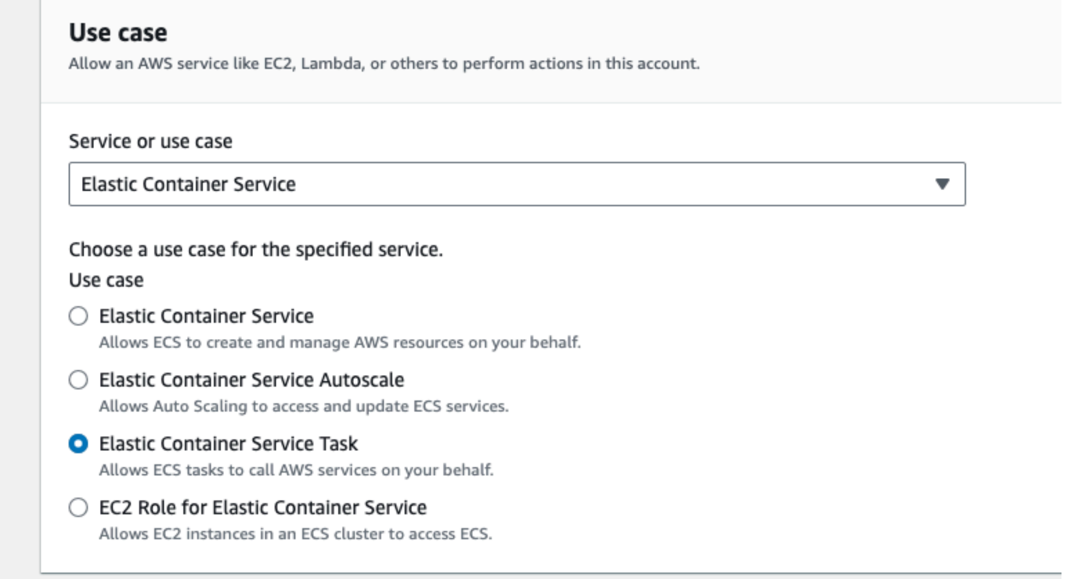
○ **AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ **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

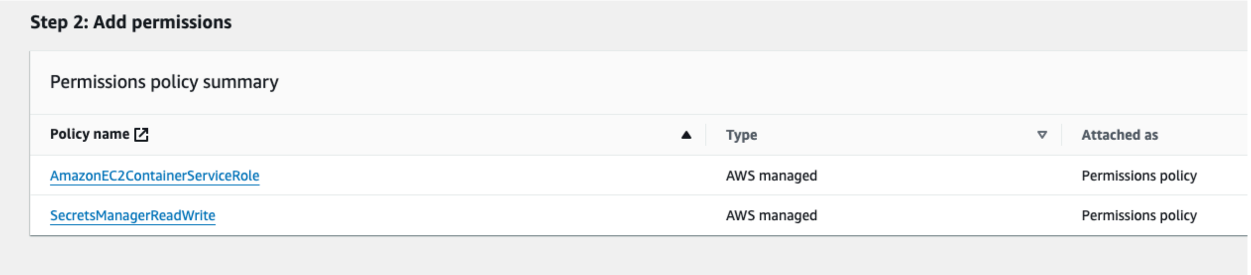
○ Elastic Container Service
Allows ECS to create and manage AWS resources on your behalf.

○ 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

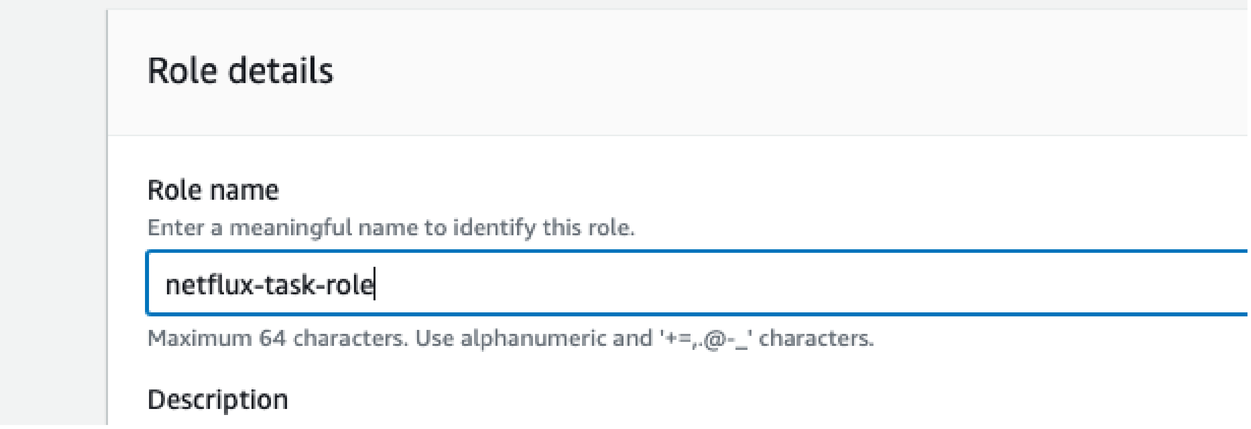


Step 2: Add permissions

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.

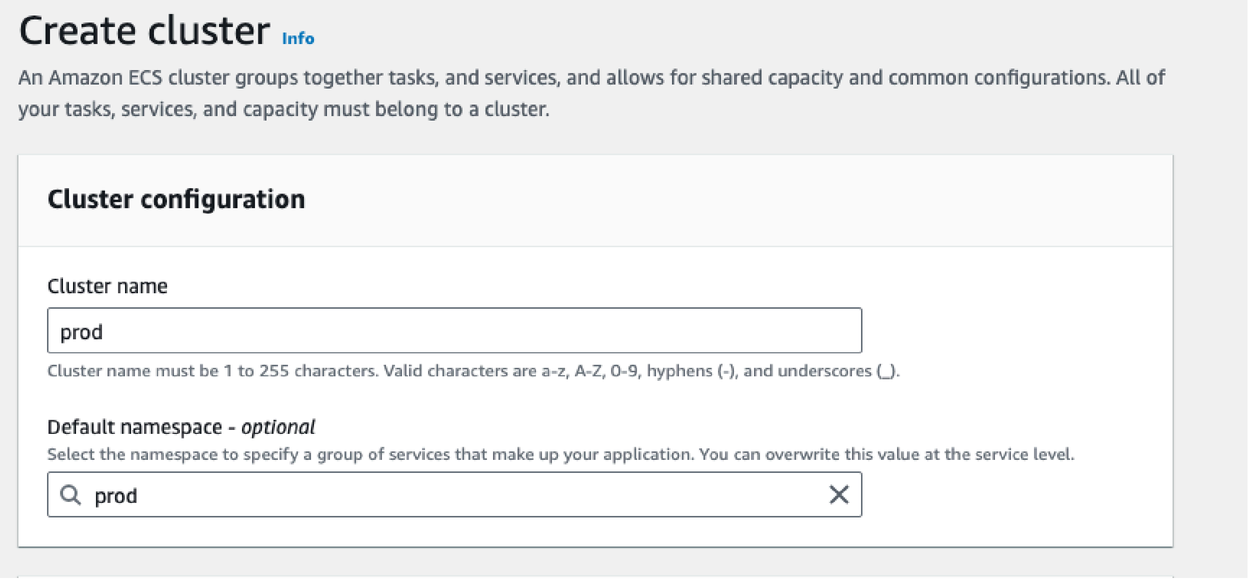
netflux-task-role

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

Description

Fargate Cluster

- Create a cluster where our tasks will be running. I call this ***prod*** 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

prod

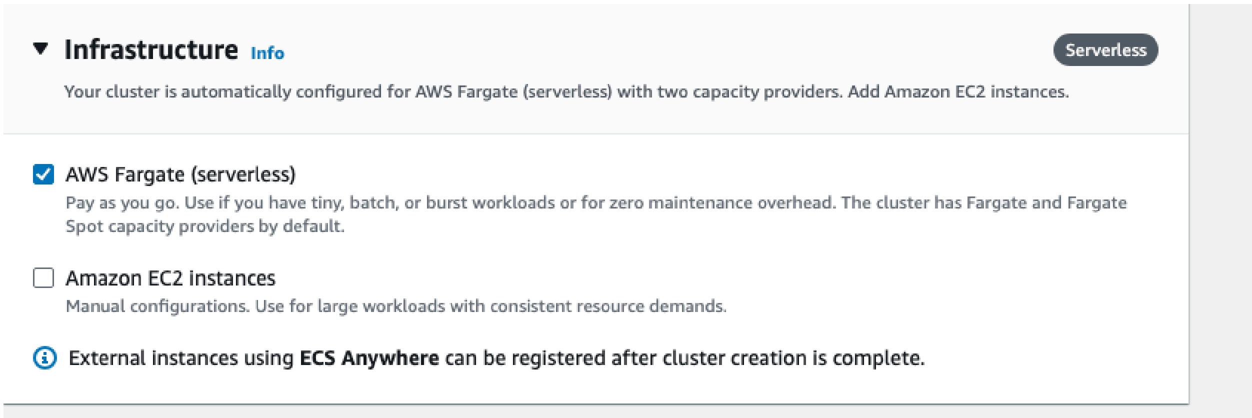
Cluster name must be 1 to 255 characters. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Default namespace - *optional*

Select the namespace to specify a group of services that make up your application. You can overwrite this value at the service level.

prod

- Select “Fargate”



Infrastructure Info

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances.

☑ AWS Fargate (serverless)

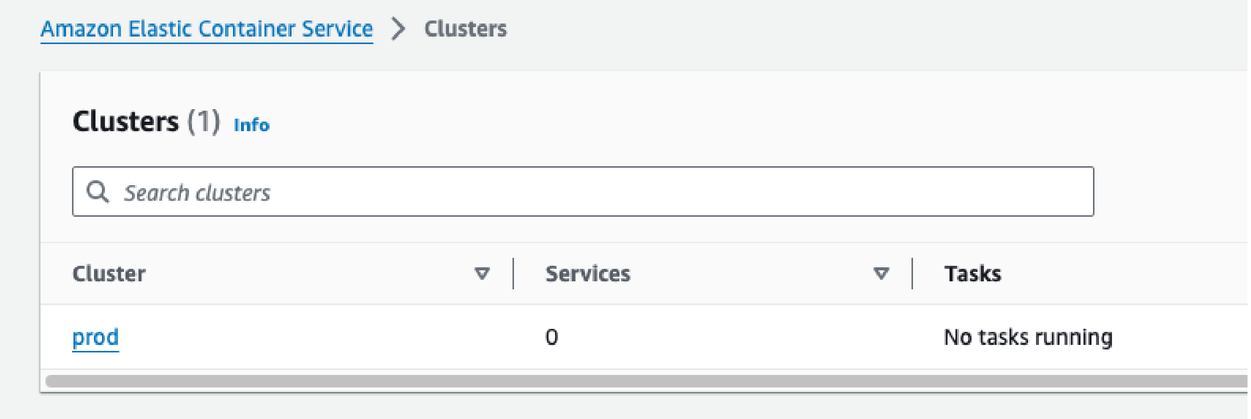
Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.

□ Amazon EC2 instances

Manual configurations. Use for large workloads with consistent resource demands.

External instances using **ECS Anywhere** can be registered after cluster creation is complete.

- It might take a minute to create the cluster



Amazon Elastic Container Service > Clusters

Clusters (1) Info

Search clusters

Cluster	Services	Tasks
prod	0	No tasks running

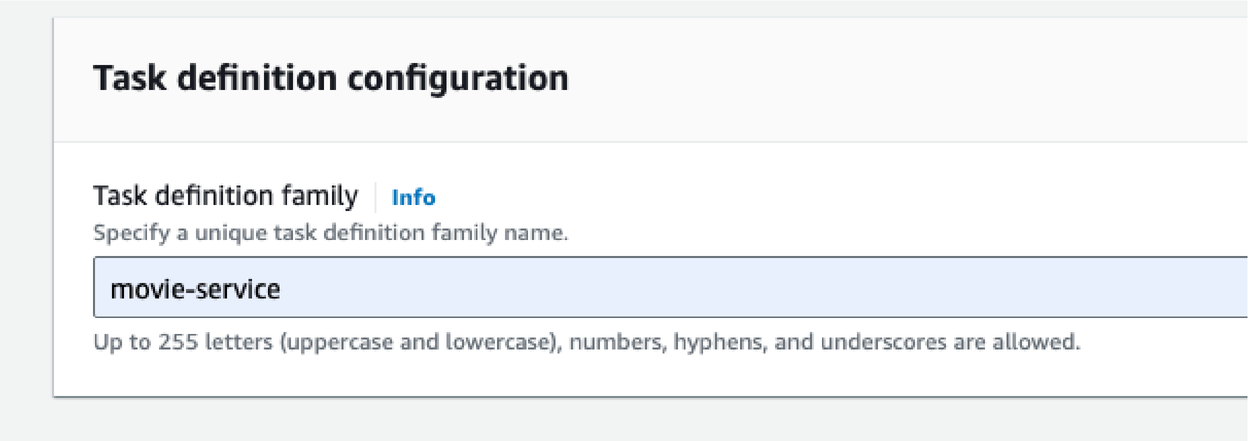
<u>Note</u><u>: If it fails for some reason, we need to go to “CloudFormation '' to clear the entry to re-create the cluster.</u>

Fargate 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

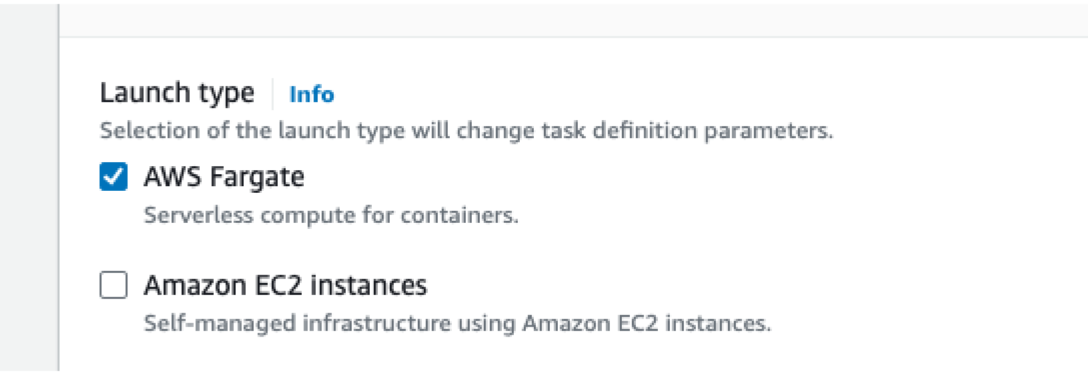
Task definition family | Info

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.

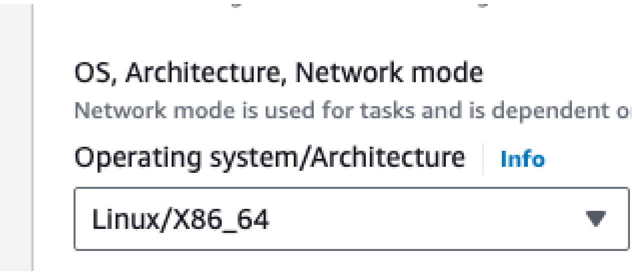
☑ AWS Fargate

Serverless compute for containers.

□ Amazon EC2 instances

Self-managed infrastructure using Amazon EC2 instances.

- CPU architecture



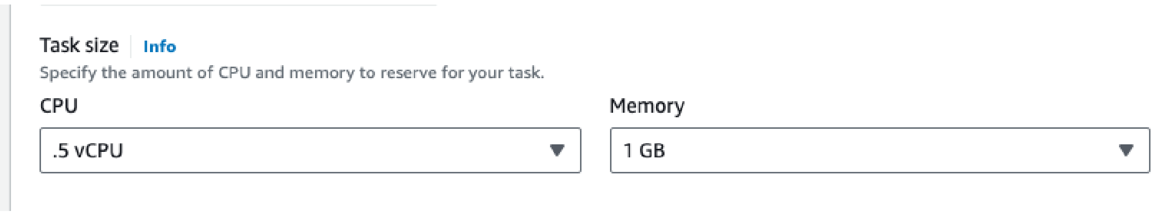
OS, Architecture, Network mode

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

Operating system/Architecture | Info

Linux/X86_64 ▼

- CPU / Memory requirements



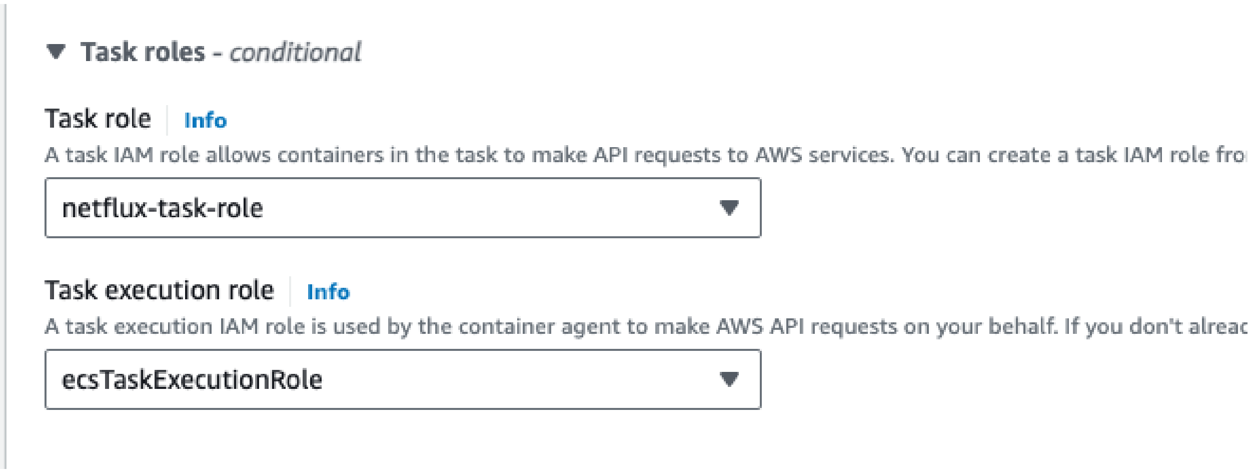
Task size

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

CPU: .5 vCPU

Memory: 1 GB

- IAM Roles



Task role

A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role fro

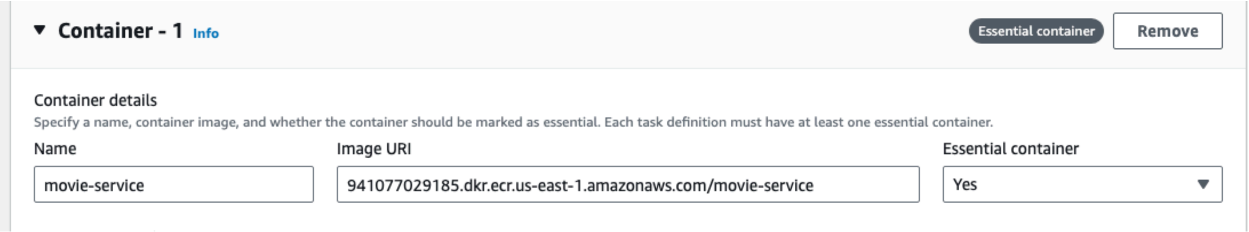
netflux-task-role

Task execution role

A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't alreac

ecsTaskExecutionRole

- docker image details

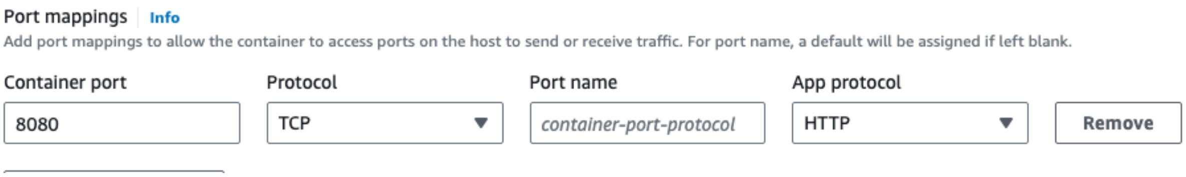


Container details

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	Image URI	Essential container
movie-service	941077029185.dkr.ecr.us-east-1.amazonaws.com/movie-service	Yes

- container port 8080

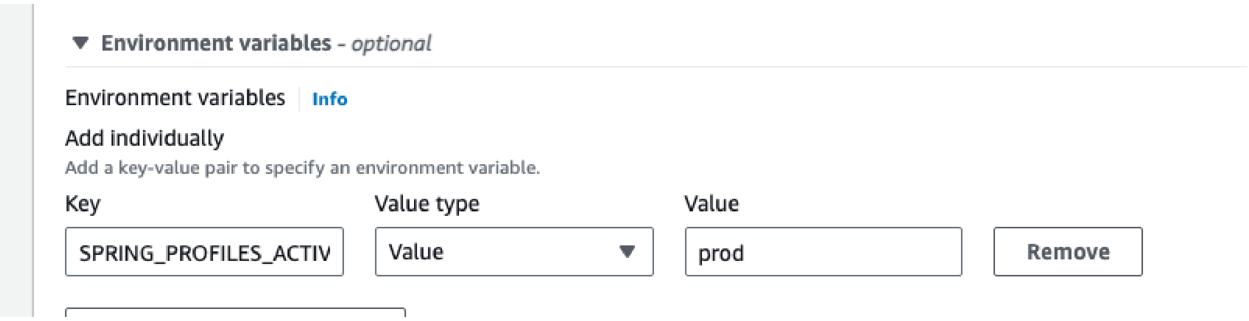


Port mappings

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

- Environment variables



Environment variables

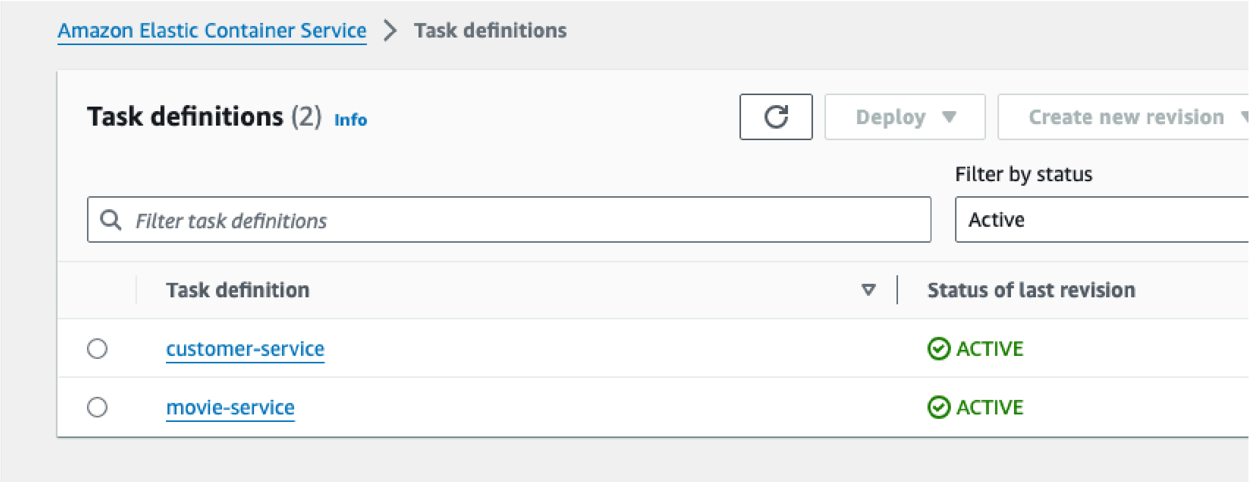
Add individually

Add a key-value pair to specify an environment variable.

Key	Value type	Value
SPRING_PROFILES_ACTIV	Value	prod

- 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

Filter by status: Active

Filter task definitions

	Task definition	Status of last revision
○	customer-service	✓ ACTIVE
○	movie-service	✓ ACTIVE

JSON reference

```
{
    "family": "customer-service",
    "containerDefinitions": [
        {
            "name": "customer-service",
            "image":
"941077029185.dkr.ecr.us-east-1.amazonaws.com/customer-service",
            "cpu": 0,
            "portMappings": [
                {
                    "name": "customer-service-8080-tcp",
                    "containerPort": 8080,
                    "hostPort": 8080,
                    "protocol": "tcp",
                    "appProtocol": "http"
                }
            ],
            "essential": true,
            "environment": [
                {
                    "name": "SPRING_PROFILES_ACTIVE",
                    "value": "prod"
                }
            ],
            "environmentFiles": [],
            "mountPoints": [],
            "volumesFrom": [],
            "ulimits": [],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/customer-service",
                    "awslogs-create-group": "true",
                    "awslogs-region": "us-east-1",
                    "awslogs-stream-prefix": "ecs"
                },
                "secretOptions": []
            },
            "systemControls": []
        }
    ],
    "taskRoleArn": "arn:aws:iam::941077029185:role/netflux-task-role",
    "executionRoleArn":
"arn:aws:iam::941077029185:role/ecsTaskExecutionRole",
    "networkMode": "awsvpc",
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "512",
    "memory": "1024",
    "runtimePlatform": {
        "cpuArchitecture": "X86_64",
        "operatingSystemFamily": "LINUX"
    }
}
```

```
{
    "family": "movie-service",
    "containerDefinitions": [
        {
            "name": "movie-service",
            "image":
```

```
"941077029185.dkr.ecr.us-east-1.amazonaws.com/movie-service",
            "cpu": 0,
            "portMappings": [
                {
                    "name": "movie-service-8080-tcp",
                    "containerPort": 8080,
                    "hostPort": 8080,
                    "protocol": "tcp",
                    "appProtocol": "http"
                }
            ],
            "essential": true,
            "environment": [
                {
                    "name": "SPRING_PROFILES_ACTIVE",
                    "value": "prod"
                }
            ],
            "environmentFiles": [],
            "mountPoints": [],
            "volumesFrom": [],
            "ulimits": [],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "/ecs/movie-service",
                    "awslogs-create-group": "true",
                    "awslogs-region": "us-east-1",
                    "awslogs-stream-prefix": "ecs"
                },
                "secretOptions": []
            },
            "systemControls": []
        }
    ],
    "taskRoleArn": "arn:aws:iam::941077029185:role/netflux-task-role",
    "executionRoleArn":
"arn:aws:iam::941077029185:role/ecsTaskExecutionRole",
    "networkMode": "awsvpc",
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "512",
    "memory": "1024",
    "runtimePlatform": {
        "cpuArchitecture": "X86_64",
        "operatingSystemFamily": "LINUX"
    }
}
```