Elastic Container Service

Elastic Container Service (ECS)

fully-managed **container** orchestration service.
Highly secure, reliable, and scalable way to run containers

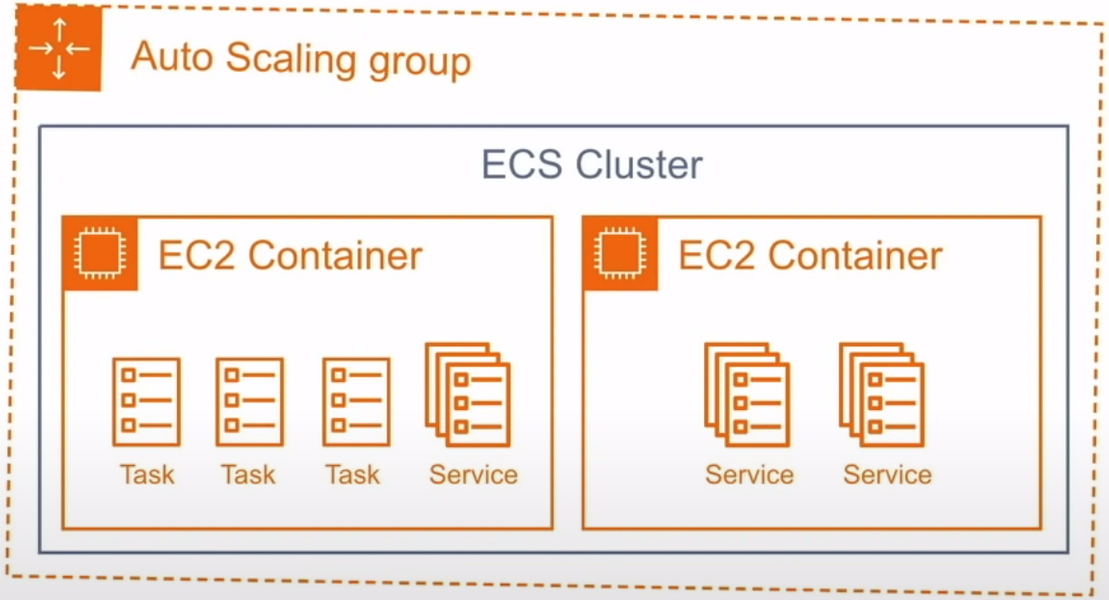
Components of ECS

Cluster
Multiple EC2 instances which will house the docker containers.

Task Definition
A JSON file that defines the configuration of (upto 10) containers you want to run

Task
Launches containers defined in Task Definition.
Tasks do not remaining running once workload is complete.

Service
Ensures tasks remaining running eg. web-app.



Container Agent
Binary on each EC2 instance which monitors, starts and stops tasks.

AWS Developer Associate

Elastic Container Service

Creating a Cluster

Creating an ECS Cluster

Create Cluster

- Use Spot or On Demand
- EC2 Instance Type
- Number of Instances
- EBS Storage Volume
- EC2 can be **Amazon Linux 2** or **Amazon Linux 1**
- Choose a VPC or create a new VPC
- Assign an IAM Role
- Option to turn on CloudWatch Container Insights
- *Choose a Key Pair

**You can SSH into an EC2 Container Instance and make changes but its not generally recommended.*

EC2 Linux + Networking

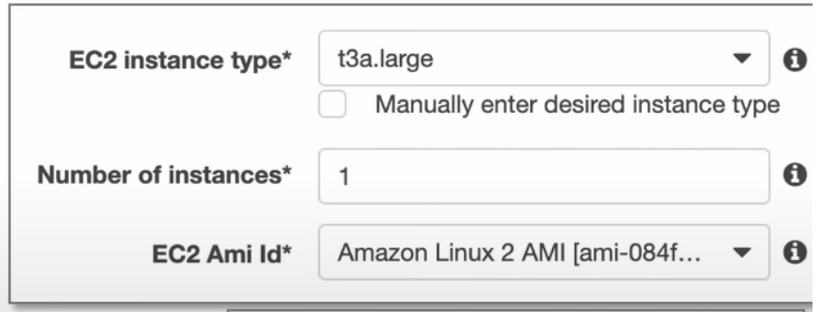
Resources to be created:

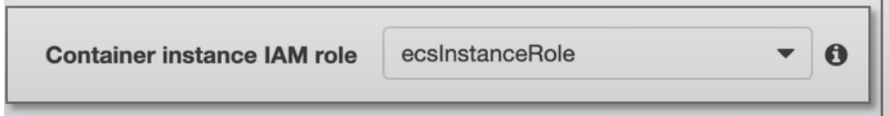
Cluster

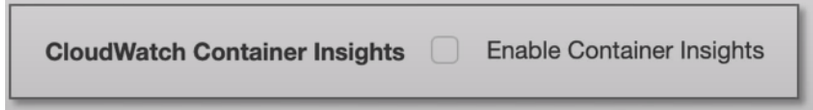
VPC

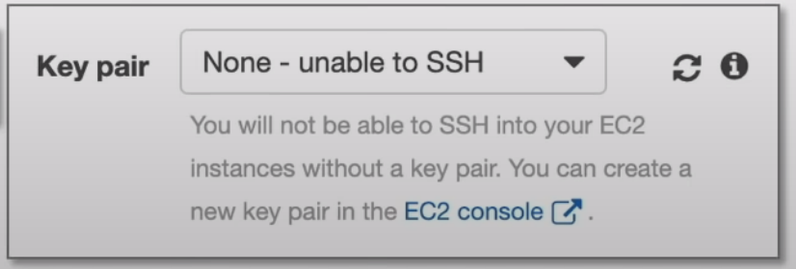
Subnets

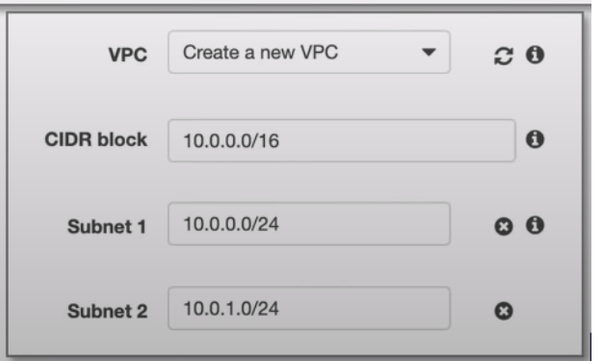
Auto Scaling group with Linux AMI











AWS Developer Associate

Elastic Container Service

Task Definitions File

Task Definition JSON File Example

Create new Task Definition

You can define multiple containers within a task definition.

The (docker) **images** can be provided either via **ECR** or an official docker repository eg. Docker Hub

You must have one **essential** container. If this container fails or stops than all other containers will be stopped.

AWS had a wizard to create Task Definitions instead having to create a file by hand

```
{
  "containerDefinitions": [
    {
      "name": "wordpress",
      "links": [
        "mysql"
      ],
      "image": "wordpress",
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80
        }
      ],
      "memory": 500,
      "cpu": 10
    },
    {
      "environment": [
        {
          "name": "MYSQL_ROOT_PASSWORD",
          "value": "password"
        }
      ],
      "name": "mysql",
      "image": "mysql",
      "cpu": 10,
      "memory": 500,
      "essential": true
    }
  ],
  "family": "hello_world"
}
```

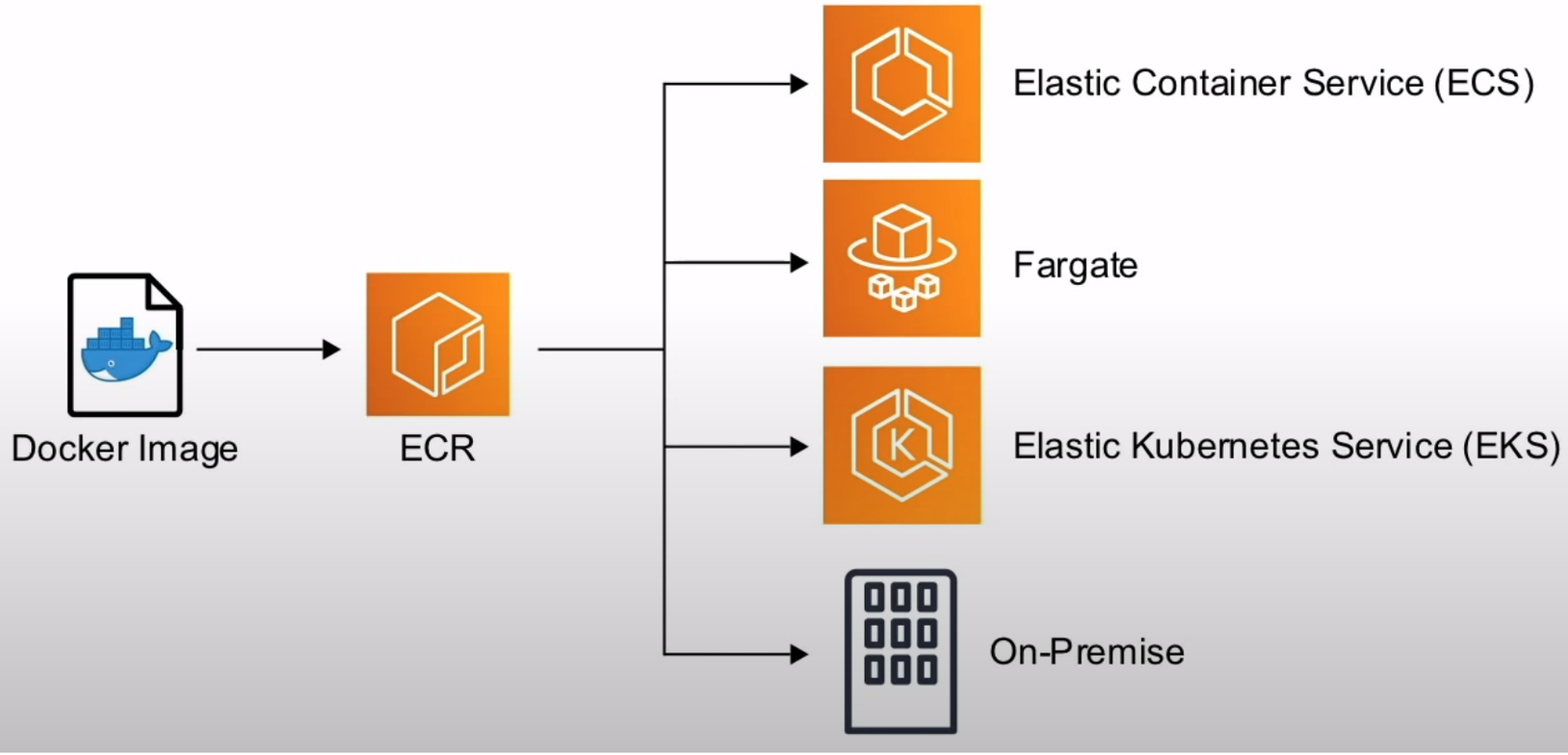
AWS Developer Associate

Elastic Container Service

Elastic Container Registry (ECR)

Elastic Container Registry (ECR)

A fully-managed Docker container registry that makes it easy for developers to **store**, **manage**, and **deploy** Docker container images.



Example 1: To create a repository

The following `create-repository` example creates a repository inside the specified namespace in the default registry for an account.

```
aws ecr create-repository \
    --repository-name project-a/nginx-web-app
```

Output:

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
nginx-web-app"
    }
}
```

Login To repository

aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 201964534042.dkr.ecr.us-east-1.amazonaws.com

WARNING! Your password will be stored unencrypted in /home/******/.docker/config.json.

Configure a credential helper to remove this warning. See

https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Create my repository

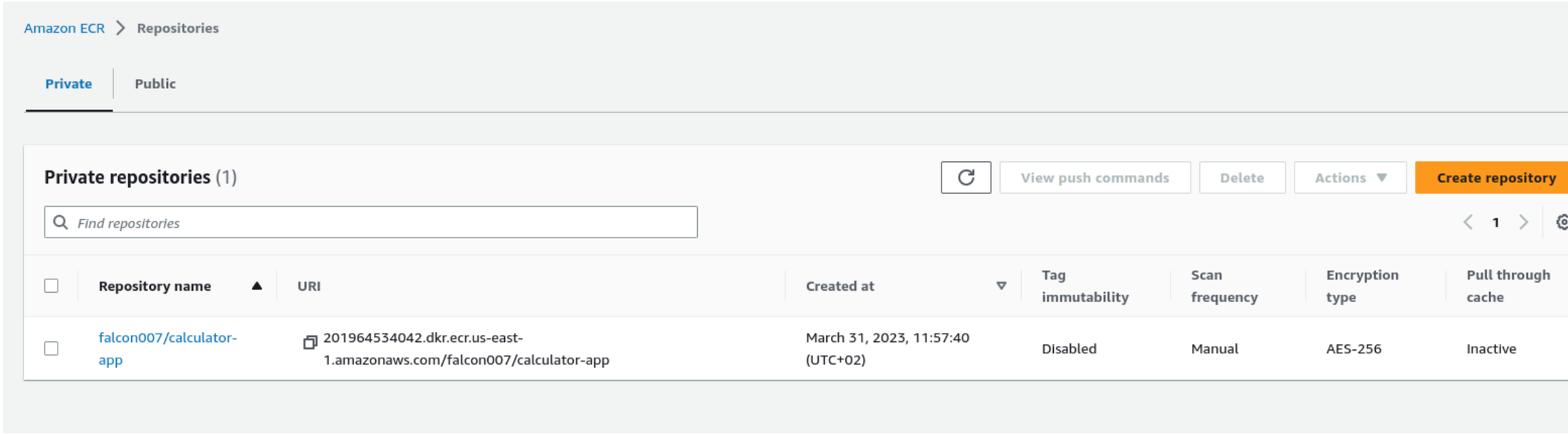
Follow the command documents:

```
docker tag falcon007/calculator-app:latest 201964534042.dkr.ecr.us-east-1.amazonaws.com/falcon007/calculator-app:latest
docker images
```

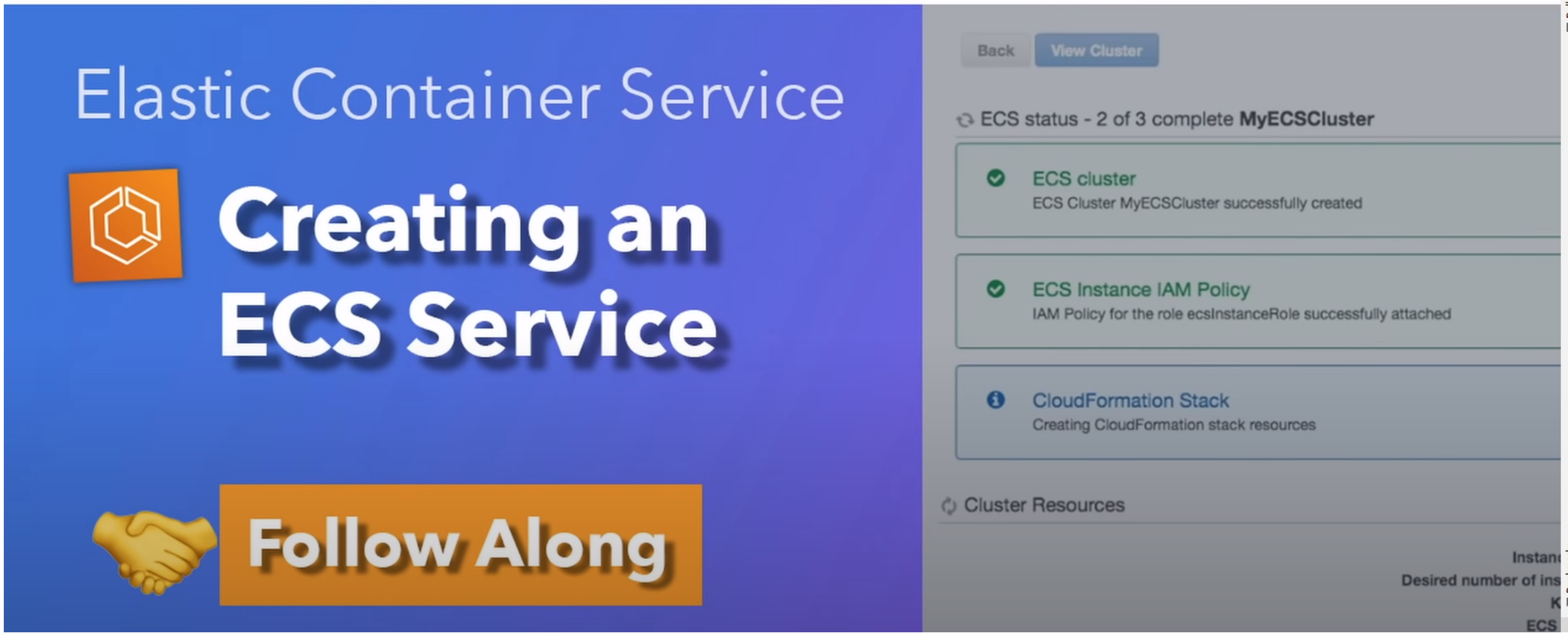
```
REPOSITORY                                                          TAG       IMAGE ID       CREATED          SIZE
201964534042.dkr.ecr.us-east-1.amazonaws.com/falcon007/calculator-app   latest    0c9bf8ebd438   52 minutes ago   496M
falcon007/calculator-app                                            latest    0c9bf8ebd438   52 minutes ago   496M
```

```
$ docker push 201964534042.dkr.ecr.us-east-1.amazonaws.com/falcon007/calculator-app:latest
```

```
The push refers to repository [201964534042.dkr.ecr.us-east-1.amazonaws.com/falcon007/calculator-app]
f0178a40a16a: Pushed
56285d9a7760: Pushed
077bff59ce57: Pushed
9cd9df9ffc97: Pushed
latest: digest: sha256:596d44169fd3b41f407468af817059132e228247887f20aaa60e9916aad62253 size: 1166
```


Amazon ECR
Repositories
Private
Public
Private repositories (1)
View push commands
Delete
Actions
Create repository
Find repositories
1
Repository name
URI
Created at
Tag immutability
Scan frequency
Encryption type
Pull through cache
falcon007/calculator-app
201964534042.dkr.ecr.us-east-1.amazonaws.com/falcon007/calculator-app
March 31, 2023, 11:57:40 (UTC+02)
Disabled
Manual
AES-256
Inactive

AWS Developer Associate



Amazon ECS Instance Role

AWS Batch compute environments are populated with Amazon ECS container instances, and they run the Amazon ECS container agent locally. The Amazon ECS container agent makes calls to various AWS APIs on your behalf, so container instances that run the agent require an IAM policy and role for these services to know that the agent belongs to you. Before you can create a compute environment and launch container instances into it, you must create an IAM role and an instance profile for those container instances to use when they are launched. This requirement applies to container instances launched with or without the Amazon ECS-optimized AMI provided by Amazon.

The Amazon ECS instance role and instance profile are automatically created for you in the console first-run experience. However, you can use the following procedure to check and see if your account already has the Amazon ECS instance role and instance profile and to attach the managed IAM policy if needed.

To check for the `ecsInstanceRole` in the IAM console

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. Search the list of roles for `ecsInstanceRole`. If the role does not exist, use the steps below to create the role.
    a. Choose **Create Role**.
    b. For **Select type of trusted entity**, choose **AWS service**. For **Choose the service that will use this role**, choose **Elastic Container Service**. For **Select your use case**, choose **EC2 Role for Elastic Container Service**.
    c. Choose **Next: Permissions**, **Next: Tags**, and **Next: Review**.
    d. For **Role Name**, type `ecsInstanceRole` and choose **Create Role**.

Previous topic: AWS Batch Service IAM Role

Next topic: Amazon EC2 Spot Fleet Role

Have a question? Try the forums

IAM > Roles

Roles (5) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Delete | Create role

Search

1

Role name	Trusted entities	Last activity
AWSCloud9SSMAccessRole	AWS Service: cloud9, and 1 more.	2 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	2 days ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked Role)	2 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-

Select trusted entity Info

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.

Use case

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

Common use cases

- (•) **EC2**
  Allows EC2 instances to call AWS services on your behalf.
- ( ) **Lambda**
  Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case ▼

Cancel Next

Add permissions Info

Permissions policies (Selected 1/828) Info

Choose one or more policies to attach to your new role.

Create policy

Filter policies by property or policy name and press enter. 7 matches

1

"ec2container" Clear filters

	Policy name	Type	Description
☐	AmazonEC2ContainerRegistryFullAccess	AWS m…	Provides administrative access to Amazon ECR resources
☐	AmazonEC2ContainerRegistryReadOnly	AWS m…	Provides read-only access to Amazon EC2 Container Registry repositories.
☐	AmazonEC2ContainerServiceEventsRole	AWS m…	Policy to enable CloudWatch Events for EC2 Container Service
☐	AmazonEC2ContainerServiceAutoscaleRole	AWS m…	Policy to enable Task Autoscaling for Amazon EC2 Container Service
☐	AmazonEC2ContainerRegistryPowerUser	AWS m…	Provides full access to Amazon EC2 Container Registry repositories, but does not allow repository deletion or policy changes.
☑	AmazonEC2ContainerServiceforEC2Role	AWS m…	Default policy for the Amazon EC2 Role for Amazon EC2 Container Service.
☐	AmazonEC2ContainerServiceRole	AWS m…	Default policy for Amazon ECS service role.

Set permissions boundary - *optional* Info

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

- (●) Create role without a permissions boundary
- ( ) Use a permissions boundary to control the maximum role permissions

Cancel Previous Next

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

ecsInstanceRole

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

Description
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

Edit

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            }
        }
    ]
}
```

IAM > Roles

Roles (6) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Delete | Create role

Search

1

Role name	Trusted entities	Last activity
AWSCloud9SSMAccessRole	AWS Service: cloud9, and 1 more.	2 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	2 days ago
AWSServiceRoleForAWSCloud9	AWS Service: cloud9 (Service-Linked Role)	2 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
ecsInstanceRole	AWS Service: ec2	-

Roles Anywhere Info

Manage

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

MyEcsCluster

There can be a maximum of 255 characters. The valid characters are letters (uppercase and lowercase), numbers, hyphens, and underscores.

Networking Info

By default tasks and services run in the default subnets for your default VPC. To use the non-default VPC, specify the VPC and subnets.

Infrastructure Info

Customized

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers.
Add Amazon EC2 instances, or external instances using ECS Anywhere.

- [x] 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.

- [x] Amazon EC2 instances
  Manual configurations. Use for large workloads with consistent resource demands.

Auto Scaling group (ASG) Info

Use Auto Scaling groups to scale the Amazon EC2 instances in the cluster.

Create new ASG

Operating system/Architecture

Choose the Windows operating system or Linux architecture for your instance.

Amazon Linux 2 (arm64)

EC2 instance type

Choose based on the workloads you plan to run on this cluster.

t2.micro

Desired capacity

Specify the number of instances to launch in your cluster.

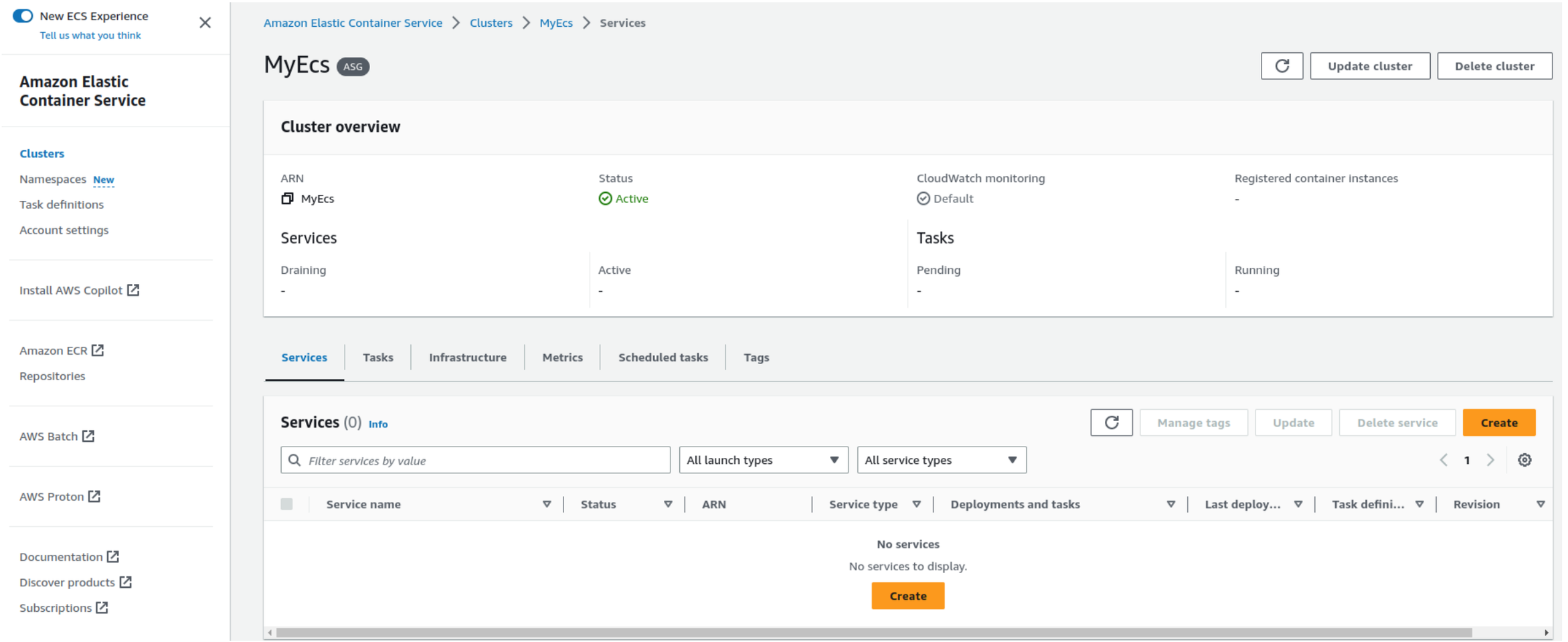
Minimum	Maximum
0	1

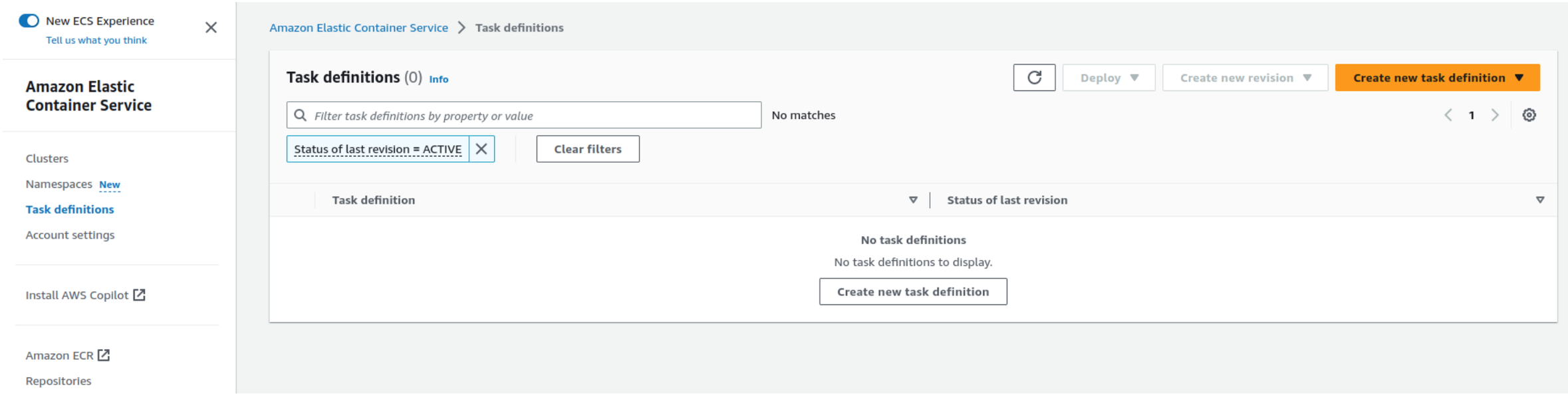
SSH Key pair

Create a key pair in the EC2 console, consisting of a private key and a public key, that you use to prove your identity when connecting to an instance.

None - unable to SSH

- [ ] External instances using ECS Anywhere
  Manual configurations. Use to add data center compute.


New ECS Experience
Tell us what you think
Amazon Elastic Container Service
Clusters
Namespaces New
Task definitions
Account settings
Install AWS Copilot
Amazon ECR
Repositories
AWS Batch
AWS Proton
Documentation
Discover products
Subscriptions
Amazon Elastic Container Service > Clusters > MyEcs > Services
MyEcs ASG
Update cluster
Delete cluster
Cluster overview
ARN
MyEcs
Status
Active
CloudWatch monitoring
Default
Registered container instances
-
Services
Draining
-
Active
-
Tasks
Pending
-
Running
-
Services
Tasks
Infrastructure
Metrics
Scheduled tasks
Tags
Services (0) Info
Manage tags
Update
Delete service
Create
Filter services by value
All launch types
All service types
1
Service name
Status
ARN
Service type
Deployments and tasks
Last deploy...
Task defini...
Revision
No services
No services to display.
Create


New ECS Experience
Tell us what you think
Amazon Elastic Container Service
Clusters
Namespaces New
Task definitions
Account settings
Install AWS Copilot
Amazon ECR
Repositories
Amazon Elastic Container Service > Task definitions
Task definitions (0) Info
Deploy
Create new revision
Create new task definition
Filter task definitions by property or value
No matches
1
Status of last revision = ACTIVE
Clear filters
Task definition
Status of last revision
No task definitions
No task definitions to display.
Create new task definition

Step 1
Configure task definition and containers

Step 2
Configure environment, storage, monitoring, and tags

Step 3
Review and create

Configure task definition and containers

Task definition configuration

Task definition family Info
Specify a unique task definition family name.

calculator

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

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	Image URI	Essential container
calculator-app	201964534042.dkr.ecr.us-east-1.amazonaws.com/falco	Yes

Port mappings Info

Add port mappings to allow the container to access ports on the host to send or receive traffic. Any changes to port mappings configuration impacts the associated service connect settings.

Host port	Container port	Protocol	Port name	App protocol	
0	80	TCP	calculator-app	HTTP	Remove

Add more port mappings

The bridge network mode is selected. By default, dynamic host port mappings are used for each container. To configure a static host port mapping, specify a host port for each container port.

▼ Environment variables - *optional* Info

Task definition successfully created

calculator:1 has been successfully created. You can use this task definition to deploy a service or run a task.

Deploy

Amazon Elastic Container Service > Task definitions > calculator > Revision 1 > Containers

calculator:1

Deploy | Actions | Create new revision

Overview Info

ARN	Status	Time created	App environment
arn:aws:ecs:us-east-1:201964534042:task-definition/calculator:1	ACTIVE	3/31/2023, 10:47:05 UTC	EC2
Task role	**Task execution role**	**Operating system/Architecture**	**Network mode**
-	ecsTaskExecutionRole	Linux/X86_64	default

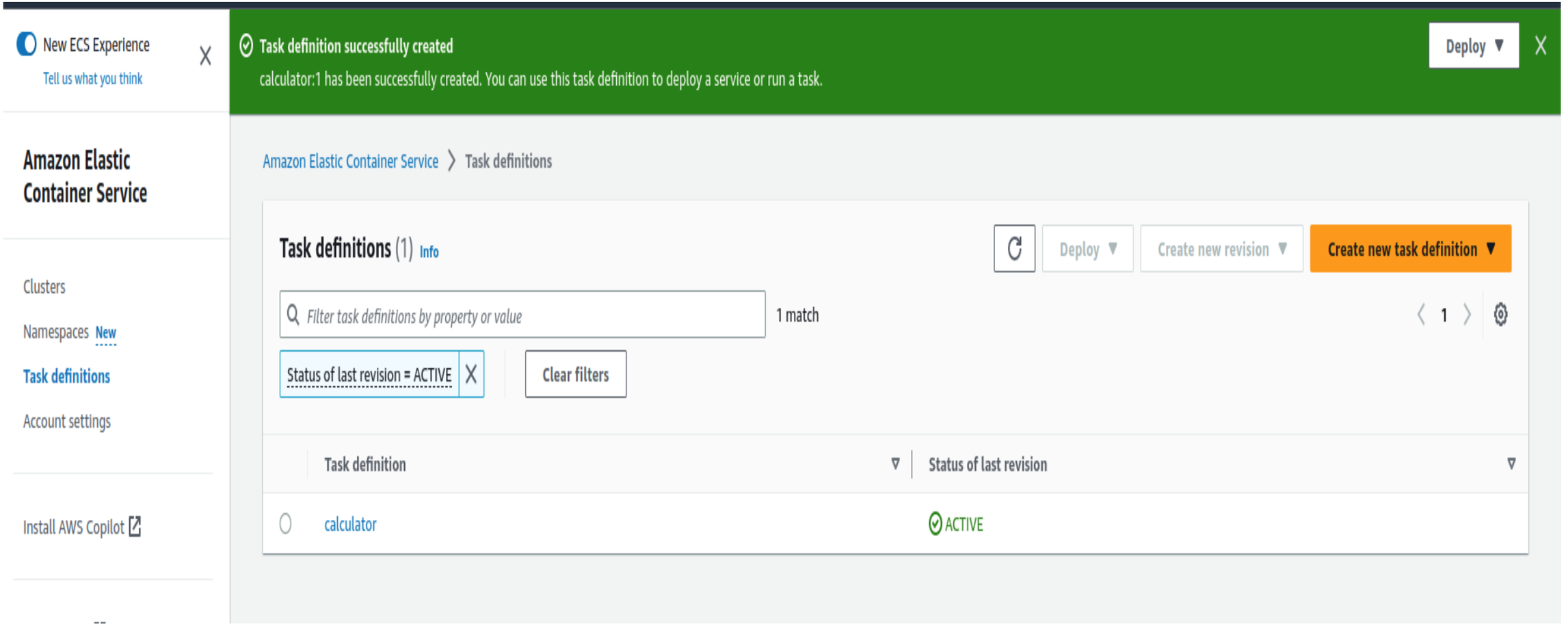
Containers | JSON | Storage | Tags

Task size

Task CPU	Task memory
1 vCPU	2 GB

Containers Info

Container name	Image	Essential	CPU	Memory	GPU
calculator-app	201964534042.dkr.ecr.us-east-...	true	0	-	-


New ECS Experience
Tell us what you think
Task definition successfully created
calculator:1 has been successfully created. You can use this task definition to deploy a service or run a task.
Deploy
Amazon Elastic Container Service
Amazon Elastic Container Service
Task definitions
Clusters
Namespaces New
Task definitions
Account settings
Task definitions (1) Info
Deploy
Create new revision
Create new task definition
Filter task definitions by property or value
1 match
1
Status of last revision = ACTIVE
Clear filters
Task definition
Status of last revision
calculator
ACTIVE
Install AWS Copilot

calculator:1 has been successfully created. You can use this task definition to deploy a service or run a task.

Amazon Elastic Container Service > Clusters > MyEcs > Services

MyEcs ASG

Update cluster | Delete cluster

Cluster overview

ARN	Status	CloudWatch monitoring	Registered container instances
MyEcs	Active	Default	-

Services

Draining	Active
-	-

Tasks

Pending	Running
-	-

Services | Tasks | Infrastructure | Metrics | Scheduled tasks | Tags

Services (0) Info

Manage tags | Update | Delete service | Create

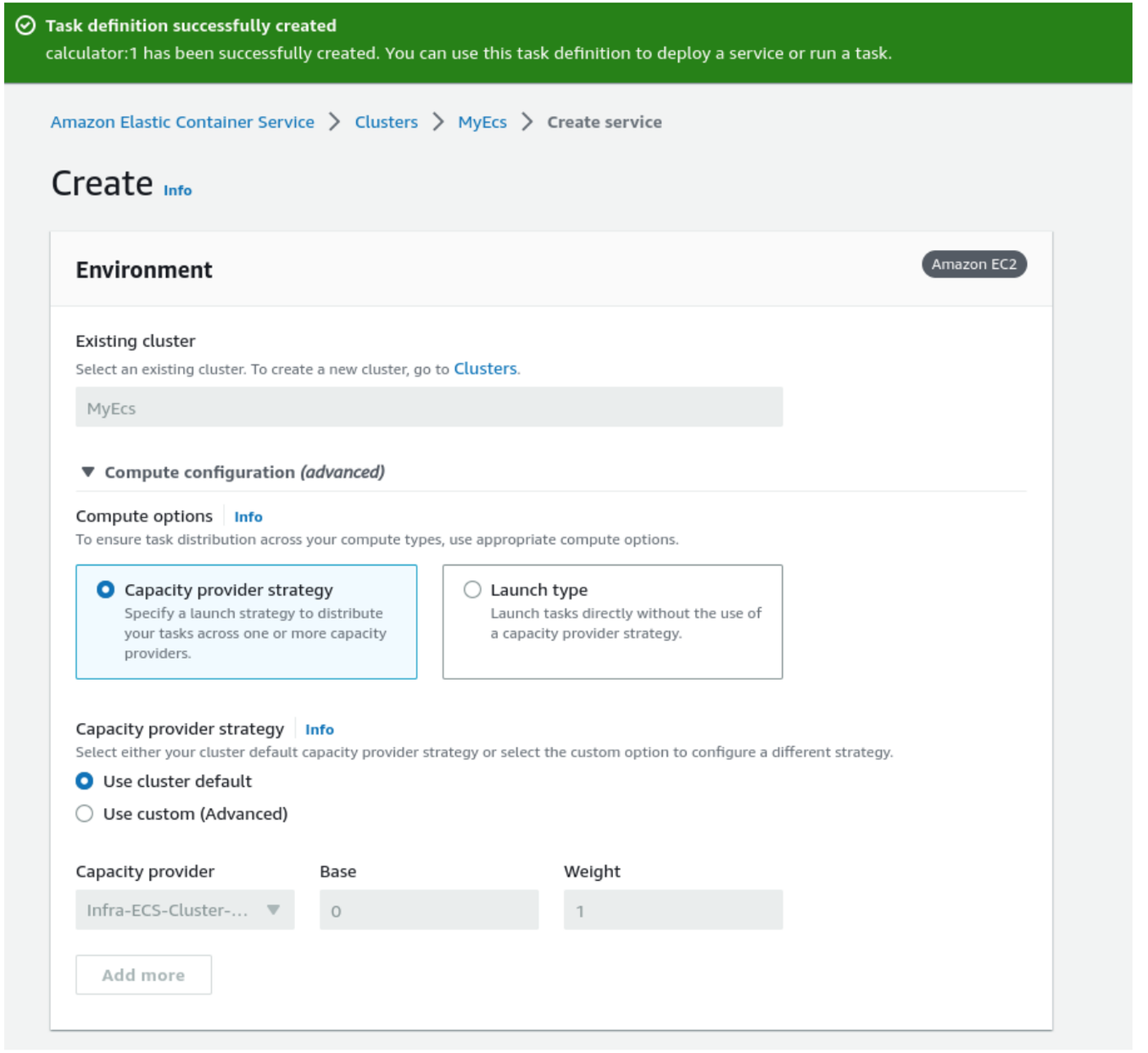
Filter services by value | All launch types | All service types | 1

Service name	Status	ARN	Service type	Deployments and tasks	Last deploy...	Task defini...	Revision

No services

No services to display.

Create


Task definition successfully created
calculator:1 has been successfully created. You can use this task definition to deploy a service or run a task.
Amazon Elastic Container Service
Clusters
MyEcs
Create service
Create
Info
Environment
Amazon EC2
Existing cluster
Select an existing cluster. To create a new cluster, go to Clusters.
MyEcs
Compute configuration (advanced)
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
Use custom (Advanced)
Capacity provider
Base
Weight
Infra-ECS-Cluster-...
0
1
Add more

Deployment configuration

Application type | Info

Specify what type of application you want to run.

- ○ **Service**
  Launch a group of tasks handling a long-running computing work that can be stopped and restarted. For example, a web application.
- ◉ **Task**
  Launch a standalone task that runs and terminates. For example, a batch job.

Task definition

Select an existing task definition. To create a new task definition, go to Task definitions.

☐ **Specify the revision manually**
Manually input the revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family	Revision
calculator	1 (LATEST)

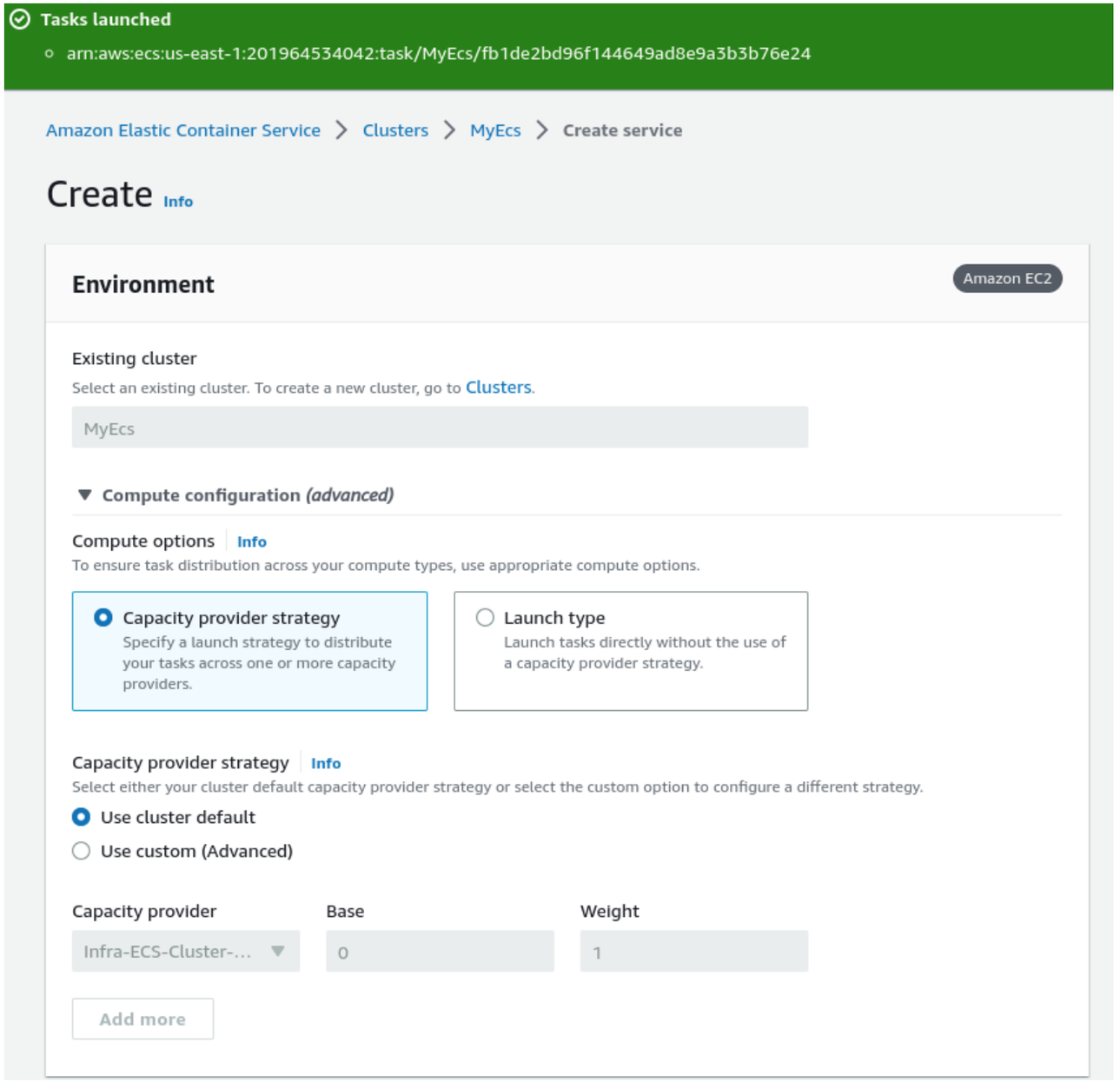
Desired tasks

Specify the number of tasks to launch.

1

Task group

All tasks with the same task group name are considered as a set when performing spread placement.


Tasks launched
arn:aws:ecs:us-east-1:201964534042:task/MyEcs/fb1de2bd96f144649ad8e9a3b3b76e24
Amazon Elastic Container Service
Clusters
MyEcs
Create service
Create
Info
Environment
Amazon EC2
Existing cluster
Select an existing cluster. To create a new cluster, go to Clusters.
MyEcs
Compute configuration (advanced)
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
Use custom (Advanced)
Capacity provider
Base
Weight
Infra-ECS-Cluster-...
0
1
Add more

AWS Developer Associate

ECS Fargate

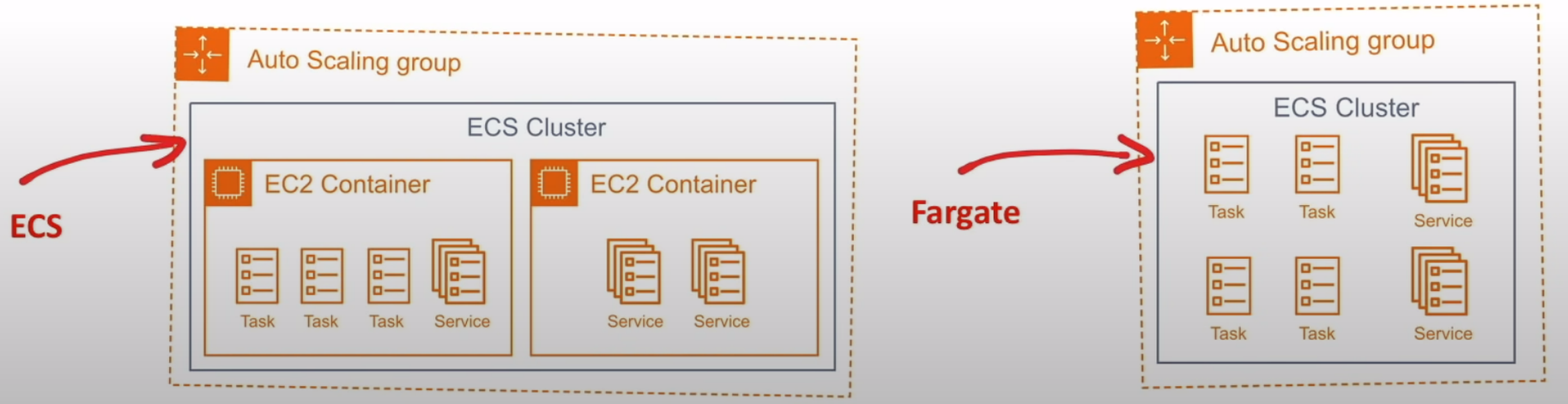
Introduction to Fargate

Fargate

Serverless containers. Don’t worry about servers.
Run containers, and pay based on duration and consumption.

Introduction to Fargate

- You can create an **empty** ECS cluster (no EC2's provisioned) and then launch Tasks as Fargate
- You **no longer have to provision**, **configure**, and **scale** clusters of EC2 instances to run containers
- You are charged for at least one minute and after its by the second.
- You pay based on duration and consumption



AWS Developer Associate

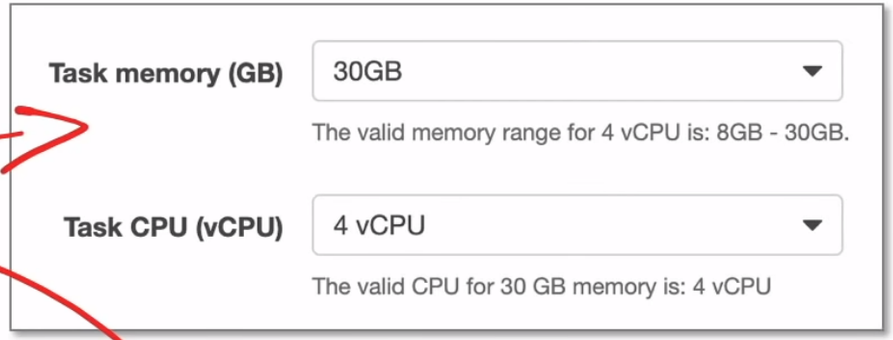
ECS Fargate

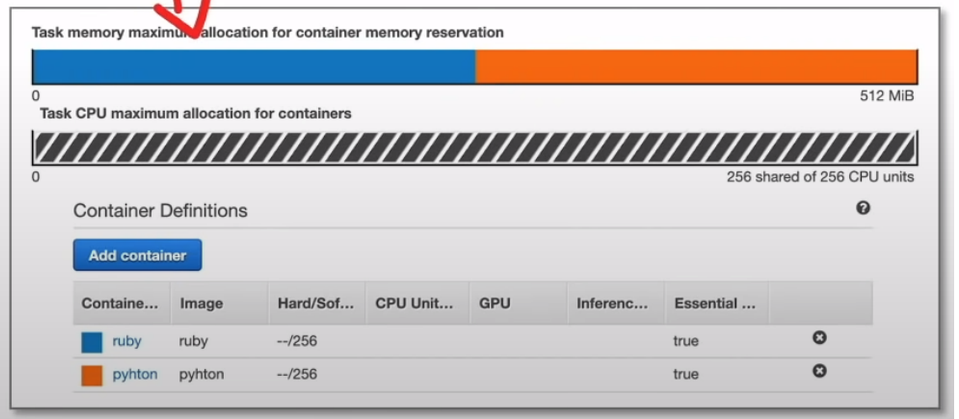
Configuring Fargate Tasks

Configuring Fargate Tasks

- In your Fargate Task Definition you **define** the **memory** and **vCPU**.
- You will then add your containers and **allocate** the memory and vCPU required for each
- When you run the Task you can choose what VPC and subnet it will run in
- **Apply a Security Group to a Task**
- Apply an IAM role to the Task

**You can apply SG and IAM role for both ECS and Fargate Tasks and Services*





AWS Developer Associate

ECS Fargate

Farvgate Vs Lambda

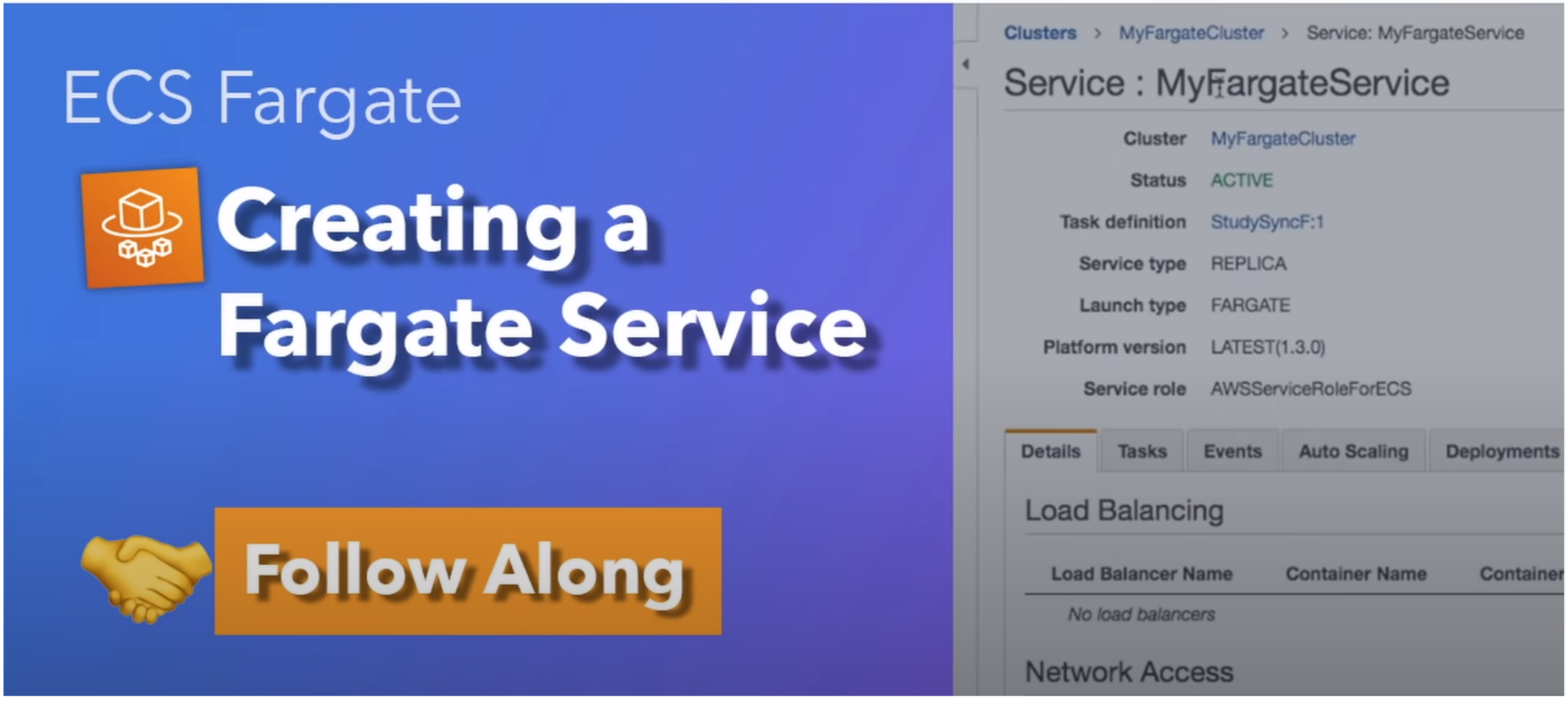
Fargate vs

Lambda

Lambda and Fargate appear very similar, but there are few key differences.

	Fargate	**Lambda**
Cold Starts	Yes (shorter)	Yes
Duration	As long as you want	**15 mins** (max)
Memory	Up to **30 GB**	Up to **3 GB**
Containers	You provide your own containers	Limited to standardize containers
Integration	More manual labour	Seamlessly integrates with other serverless services.
Pricing	Pay at least 1 min and every additional second.	Pay per 100ms

AWS Developer Associate



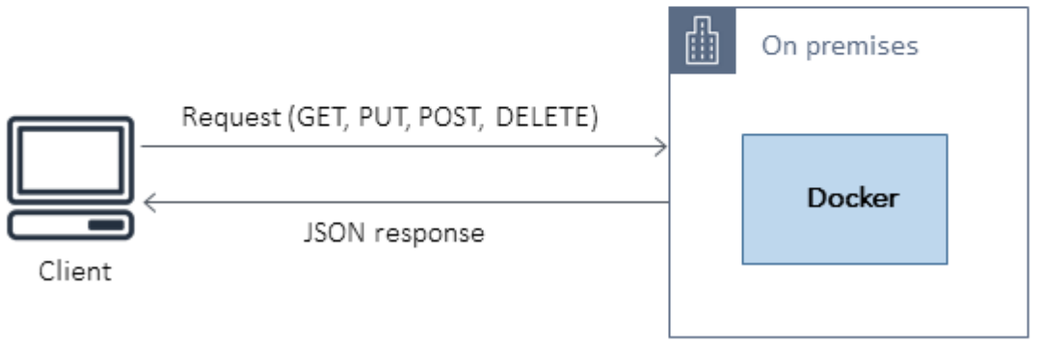
Deploy Java microservices on Amazon ECS using AWS Fargate

Architecture

Source technology stack

- Java microservices (for example, implemented in Spring Boot) and deployed on Docker

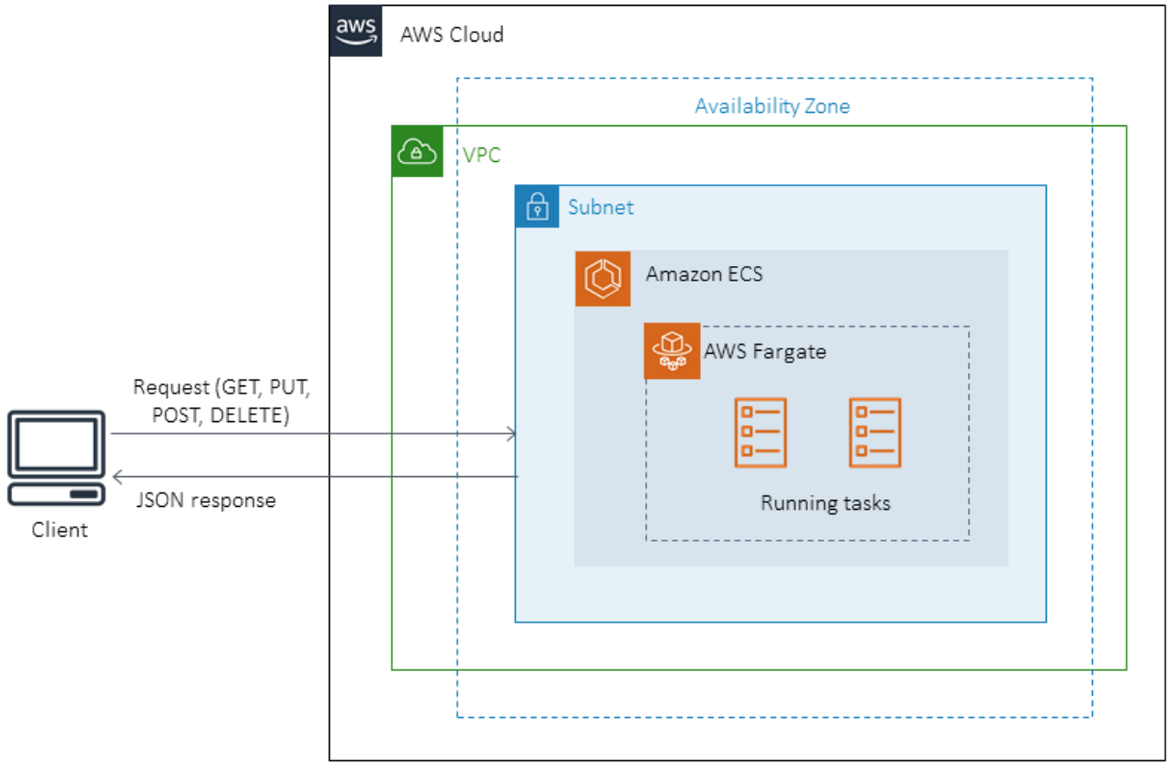
Source architecture



Target technology stack

- An Amazon ECS cluster that hosts each microservice by using Fargate
- A VPC network to host the Amazon ECS cluster and associated security groups
- A cluster/task definition for each microservice that spins up containers by using Fargate

Target architecture



Tools

Tools

- Amazon ECS eliminates the need to install and operate your own container orchestration software, manage and scale a cluster of virtual machines, or schedule containers on those virtual machines.
- AWS Fargate helps you run containers without needing to manage servers or Amazon Elastic Compute Cloud (Amazon EC2) instances. It's used in conjunction with Amazon Elastic Container Service (Amazon ECS).
- Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called *containers* that have everything the software needs to run, including libraries, system tools, code, and runtime.

Docker code

The following Dockerfile specifies the Java Development Kit (JDK) version that is used, where the Java archive (JAR) file exists, the port number that is exposed, and the entry point for the application.

```
FROM openjdk:11
ADD target/Spring-docker.jar Spring-docker.jar
EXPOSE 8080
ENTRYPOINT ["java","-jar","Spring-docker.jar"]
```

Epics

Create new task definitions

Task	Description	Skills required
Create a task definition.	Running a Docker container in Amazon ECS requires a task definition. Open the Amazon ECS console at https://console.aws.amazon.com/ecs/, choose **Task definitions**, and then create a new task definition. For more information, see the Amazon ECS documentation.	AWS systems administrator, App developer
Choose launch type.	Choose **Fargate** as the launch type.	AWS systems administrator, App developer
Configure the task.	Define a task name and configure the application with the appropriate amount of task memory and CPU.	AWS systems administrator, App developer
Define the container.	Specify the container name. For the image, enter the Docker site name, the repository name, and the tag name of the Docker image (`docker.io/sample-repo/sample-application:sample-tag-name`). Set memory limits for the application, and set port mappings (`8080, 80`) for the allowed ports.	AWS systems administrator, App developer
Create the task.	When the task and container configurations are in place, create the task. For detailed instructions, see the links in the *Related resources* section.	AWS systems administrator, App developer

Configure Task

Task	Description	Skills required
Create a task.	Inside the cluster, choose **Run new task**.	AWS systems administrator, App developer
Choose launch type.	Choose **Fargate** as the launch type.	AWS systems administrator, App developer
Choose task definition, revision, and platform version.	Choose the task that you want to run, the revision of the task definition, and the platform version.	AWS systems administrator, App developer
Select the cluster.	Choose the cluster where you want to run the task from.	AWS systems administrator, App developer
Specify the number of tasks.	Configure the number of tasks that should run. If you're launching with two or more tasks, a load balancer is required to distribute the traffic among the tasks.	AWS systems administrator, App developer
Specify the task group.	(Optional) Specify a task group name to identify a set of related tasks as a task group.	AWS systems administrator, App developer
Configure the cluster VPC, subnets, and security groups.	Configure the cluster VPC and the subnets on which you want to deploy the application. Create or update security groups (HTTP, HTTPS, and port 8080) to provide access to inbound and outbound connections.	AWS systems administrator, App developer
Configure public IP settings.	Enable or disable the public IP, depending on whether you want to use a public IP address for Fargate tasks. The default, recommended option is **Enabled**.	AWS systems administrator, App developer
Review settings and create the task	Review your settings, and then choose **Run Task**.	AWS systems administrator, App developer

Cut over

Task	Description	Skills required
Copy the application URL.	When the task status has been updated to *Running*, select the task. In the Networking section, copy the public IP.	AWS systems administrator, App developer
Test your application.	In your browser, enter the public IP to test the application.	AWS systems administrator, App developer

ECS and Fargate Cheat Sheet

- **Elastic Container Service (ECS)** is fully-managed container orchestration service. Highly secure, reliable, and scalable way to run containers
- Components of ECS
  - **Cluster** Multiple EC2 instances which will house the docker containers.
  - **Task Definition** A JSON file that defines the configuration of (upto 10) containers you want to run
  - **Task** Launches containers defined in Task Definition. Tasks do not remaining running once workload is complete.
  - **Service** Ensures tasks remaining running eg. web-app.
  - **Container Agent** Binary on each EC2 instance which monitors, starts and stops tasks

Elastic Container Registry (ECR) A fully-managed Docker container registry that makes it easy for developers to **store**, **manage**, and **deploy** Docker container images.

- **Fargate** is **Serverless containers.** Don't worry about servers. Run containers, and pay based on duration and consumption
  - Fargate has **Cold Starts** so if this is an issue for you then use ECS.
  - **Duration** As long as you want
  - **Memory** Up to 30 GB
  - **Pricing** Pay at least 1 min and every additional second