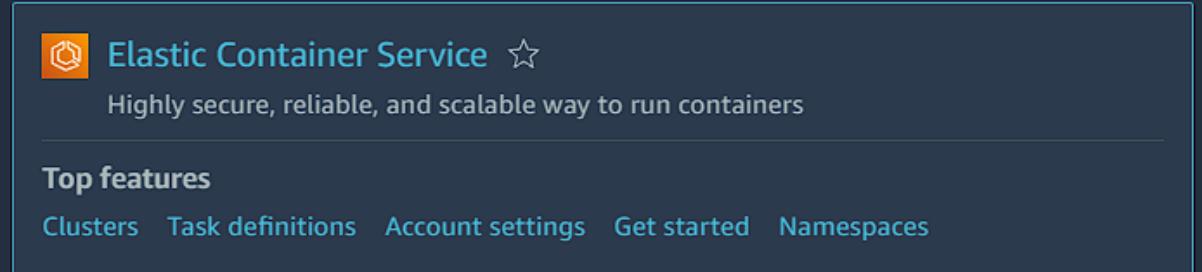


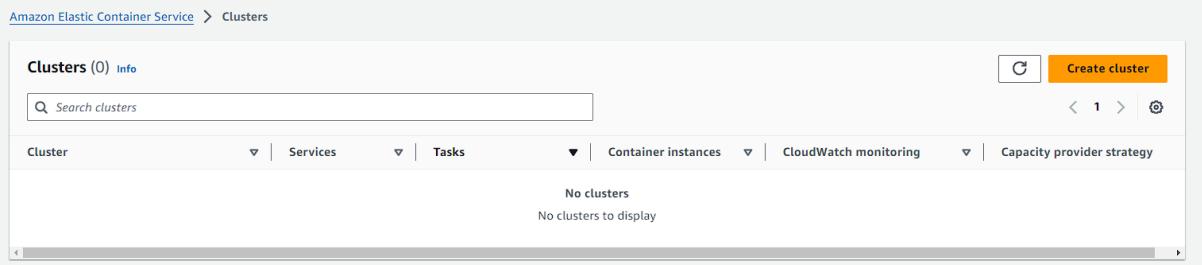
Elastic Container Service

1. In this lab you are going to learn about ECS hands on.
2. The prerequisites used in this lab are from previous lab.
3. Now you need to navigate to ECS. Choose this service accordingly.



The screenshot shows the AWS Elastic Container Service (ECS) homepage. At the top, there's a large orange hexagonal icon with a white smiley face. To its right, the text "Elastic Container Service" is displayed in a large, bold, black font, followed by a yellow star icon. Below this, a sub-headline reads "Highly secure, reliable, and scalable way to run containers". Underneath, a section titled "Top features" is shown with five links: "Clusters", "Task definitions", "Account settings", "Get started", and "Namespaces".

4. Now you need to create a cluster. So, click on create cluster.



The screenshot shows the "Clusters" page within the AWS ECS console. At the top left, it says "Clusters (0) Info". On the right, there's a prominent orange "Create cluster" button. Below the button is a search bar labeled "Search clusters". The main area has a table header with columns: Cluster, Services, Tasks, Container instances, CloudWatch monitoring, and Capacity provider strategy. A message at the bottom center states "No clusters" and "No clusters to display".

5. Now you need to give it a name here and, in the infrastructure, you need to select AWS Fargate (serverless).
6. Then scroll down to the bottom and create your cluster.

Cluster configuration

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

▼ Infrastructure Info Serverless

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

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
Manual configurations. Use to add data center compute.

7. Once you have the cluster in place. Then you need to go to Task Definition.
8. There you need to click on create task definition.

Clusters

Namespaces

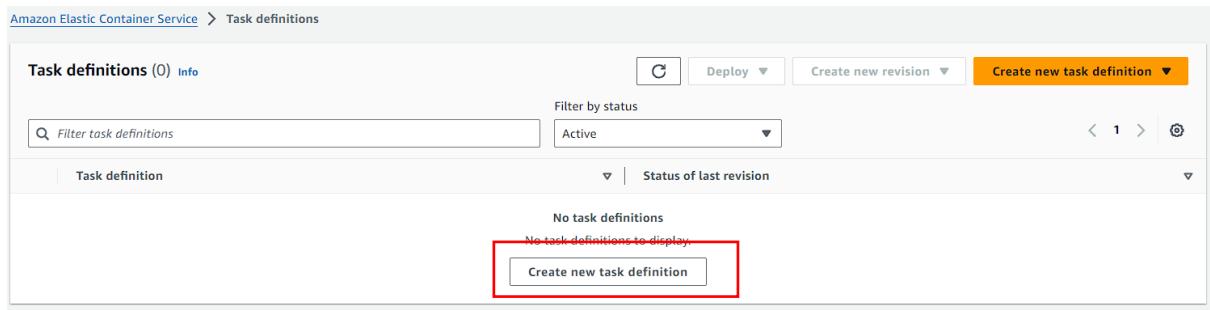
Task definitions

Account settings

Amazon Elastic Container Service > Clusters

Clusters (1) Info Create cluster

Cluster	Services	Tasks	Container instances	CloudWatch monitoring	Capacity provider strategy
demo-cluster	0	No tasks running	0 EC2	Default	No default found



9. First give it a name then scroll down and select the infrastructure as AWS Fargate and keep the rest things to default.

The screenshot shows the 'Task definition configuration' page. In the 'Task definition family' section, the input field contains 'demo-task', which is highlighted with a blue border. Below the input field, a note says 'Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.'

In the 'Infrastructure requirements' section, the 'Launch type' dropdown is set to 'AWS Fargate', indicated by a checked checkbox. The description below states 'Serverless compute for containers.'

The 'OS, Architecture, Network mode' section includes a 'Operating system/Architecture' dropdown set to 'Linux/X86_64'.

The 'Network mode' dropdown is set to 'awsvpc'.

The 'Task size' section shows 'CPU' set to '1 vCPU' and 'Memory' set to '3 GB'.

10. Now you need to create a task role which will be created in IAM and the task execution role will be create itself.

▼ 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](#).

-

▼

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.

Create new role

▼

▼ Task placement - *optional*

i Task placement constraints are not supported for AWS Fargate launch type.

11. Now navigate to IAM in another tab. In the use case you need to select the service as Elastic container service and the use case as ECS Task.
12. For the permission add SQS full access for it. Then just simply create your role.

The screenshot shows the AWS IAM Role creation wizard. In the first step, 'Use case', the 'Service or use case' dropdown is set to 'Elastic Container Service'. Under 'Use case', the 'Elastic Container Service Task' option is selected, which allows ECS tasks to call AWS services on behalf of the user. In the second step, 'Permissions', a search bar with 'sq' is used to filter policies. The 'AmazonSQSFullAccess' policy is selected, and the table shows three matches:

Policy name	Type	Description
AmazonSQSFullAccess	AWS managed	Provides full access to Amazon SQS via t...
AmazonSQSReadOnlyAccess	AWS managed	Provides read only access to Amazon SQ...
AWSLambdaSQSQueueExecutionRole	AWS managed	Provides receive message, delete messag...

13. Now come back to ECS and create you task definition keep the settings same as they were. This time just select your role.
14. Now in the container 1 you need to specify the name of your container and give the Image URI from your ECR. In ECR open you image you will your URI.

The screenshot shows the 'Container - 1' configuration screen. The 'Name' field is set to 'sendmessages', the 'Image URI' is '878893308172.dkr.ecr.ap-south-1.amazonaws.com/sendmessages:latest', and the 'Essential container' dropdown is set to 'Yes'.

15. Once your task definition is created from click on deploy and then on run task.

The screenshot shows the AWS CloudWatch Task Definition Overview page for a task named 'demotaskecs:1'. At the top right, there are three buttons: 'Deploy ▲', 'Actions ▼', and 'Create new revision ▾'. The 'Run task' button is highlighted with a red box. Below these buttons, there are two dropdown menus: 'Create service' and 'Update service'. The main content area displays task details such as ARN, Status (ACTIVE), Time created (2024-02-15T08:55:12.001Z), App environment (FARGATE), Task role (ECS-SQS), Task execution role (ecsTaskExecutionRole), Operating system/Architecture (Linux/X86_64), and Network mode (awsvpc). Below this, there are tabs for 'Containers', 'JSON', 'Task placement', 'Volumes (0)', 'Requires attributes', and 'Tags'. Under the 'Task size' section, it shows Task CPU (1024 units (1 vCPU)), Task memory (3072 MiB (3 GB)), Task CPU maximum allocation for containers (CPU (unit) slider), and Task memory maximum allocation for container memory reservation (Memory (MiB) slider).

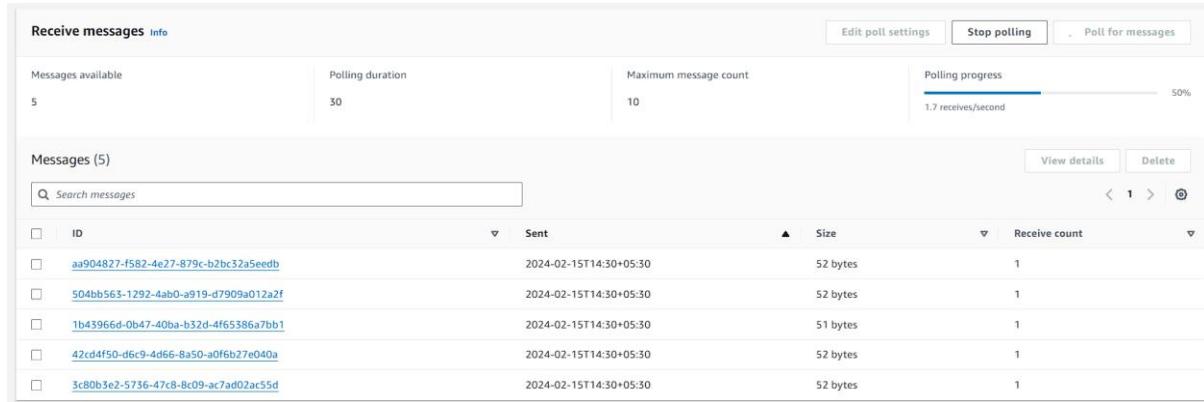
16. In the run task it will ask you to create an environment in that just select you existing cluster and click on create.

The screenshot shows the 'Run task' configuration dialog. The 'Environment' tab is selected. It features an 'Existing cluster' dropdown set to 'demo-cluster', a 'Create a new cluster' button, and a 'Compute configuration (advanced)' section. This section includes 'Compute options' and a note about ensuring task distribution across compute types. It contains two radio buttons: 'Capacity provider strategy' (selected) and 'Launch type'. The 'Capacity provider strategy' section includes a note about specifying a launch strategy to distribute tasks across one or more capacity providers. The 'Launch type' section includes a note about launching tasks directly without using a capacity provider strategy. Below this is a 'Capacity provider strategy' section with a note about selecting either the cluster default or custom options. It shows two radio buttons: 'Use cluster default' (unchecked) and 'Use custom (Advanced)' (checked).

17. You can see your task is in provisioning state.

The screenshot shows the AWS CloudWatch Tasks page. The 'Tasks (1)' section displays a table with one row. The columns include Task, Last status, Desired st..., Task d..., Health sta..., Started at, Container instan..., Launch type, Platform ..., and CF. The first column shows a checkbox and the text '36db7...'. The second column shows a status icon and 'Provisioning'. The third column shows a status icon and 'Running'. The fourth column shows a status icon and 'demotask...'. The fifth column shows a status icon and 'Unknown'. The sixth column shows a status icon and '-'. The seventh column shows a status icon and 'FARGATE'. The eighth column shows a status icon and '1.4.0'. The ninth column shows a status icon and '1'. At the top of the table, there are filters for 'Filter desired status' (set to 'Running') and 'Filter launch type' (set to 'Any launch type'). There are also buttons for 'Stop ▾' and 'Run new task'.

18. Once your task is complete navigate to SQS and there click on start polling you will see your messages there.



The screenshot shows the 'Receive messages' interface in the AWS SQS console. At the top, there are buttons for 'Edit poll settings', 'Stop polling', and 'Poll for messages'. Below this, there are four input fields: 'Messages available' (5), 'Polling duration' (30), 'Maximum message count' (10), and a progress bar for 'Polling progress' (1.7 receives/second, 50%). A table below lists the five messages in the queue:

ID	Sent	Size	Receive count
aa904827-f582-4e27-879c-b2bc32a5eedb	2024-02-15T14:30+05:30	52 bytes	1
504bb563-1292-4ab0-a919-d7909a012a2f	2024-02-15T14:30+05:30	52 bytes	1
1b43966d-0b47-40ba-b32d-4f65386a7bb1	2024-02-15T14:30+05:30	51 bytes	1
42cd4f50-d6c9-4d66-8a50-a0f6b27e040a	2024-02-15T14:30+05:30	52 bytes	1
3c80b5e2-5736-47c8-8c09-ac7ad02ac55d	2024-02-15T14:30+05:30	52 bytes	1

19. So, for the clean up first delete your task definition then delete your cluster.