Introduction

An ECS service is a mechanism that allows ECS to run and maintain a specified number of instances of a task definition. If any tasks or container instances should fail or stop, the ECS service scheduler launches another instance to replace it. This is similar to Auto Scaling in that it maintains a desired number of instances, but it does not scale instances up or down based on CloudWatch alarms or other Auto Scaling mechanisms. Services behind a load balancer provide a relatively seamless way to maintain a certain amount of resources while keeping a single application reference point.

In this lab step, you will create two services, one for your blue application and one for the green application.

Instructions

1. In the Amazon ECS console, in the left-hand menu, click **Clusters**, and in the table, click **ecslab-cluster**:

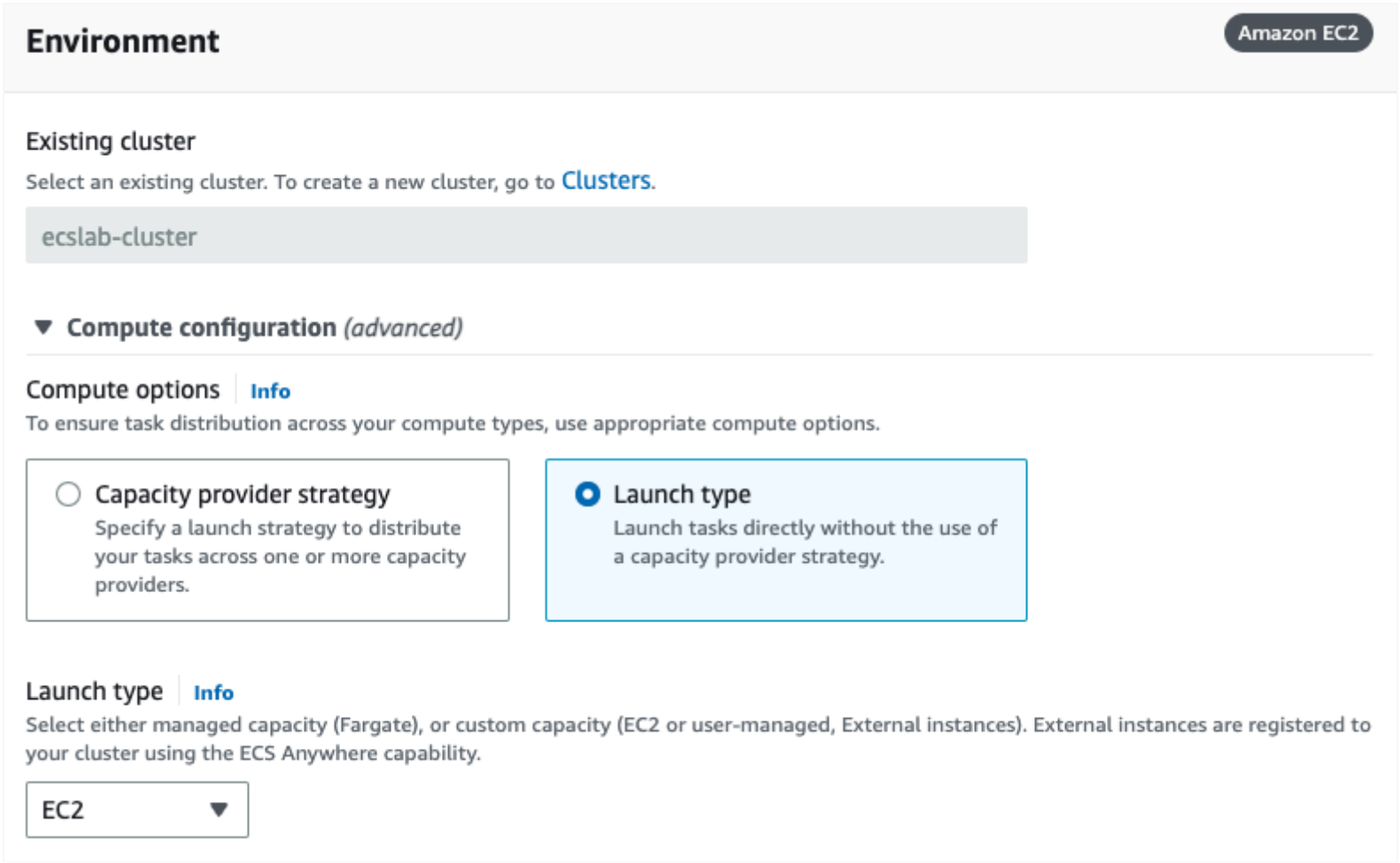
The **Cluster overview** page will load.

2. To begin creating a new Amazon ECS service, in the **Services** tab at the bottom of the page, click **Create**:

A form titled **Create** will load.

3. Enter and select the following to configure a service for your blue application:

- **Environment**:
  - **Compute Options**: Launch type
  - **Launch Type**: EC2
- **Deployment configuration**:
  - **Application type**: Ensure **Service** is selected
  - **Family**: Select **ecslab-blue-taskdef**
  - **Revision**: Ensure the selected **Revision** is **LATEST**
  - **Service name**: Enter *ecslab-blue-service*
  - **Desired tasks***:* Replace **1** with *2*


Environment
Amazon EC2
Existing cluster
Select an existing cluster. To create a new cluster, go to Clusters.
ecslab-cluster
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.
Launch type Info
Select either managed capacity (Fargate), or custom capacity (EC2 or user-managed, External instances). External instances are registered to your cluster using the ECS Anywhere capability.
EC2

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

ecslab-blue-taskdef

Revision

2 (LATEST)

Service name

Assign a unique name for this service.

ecslab-blue-service

Service type Info

Specify the service type that the service scheduler will follow.

- (•) **Replica**
  Place and maintain a desired number of tasks across your cluster.
- ( ) **Daemon**
  Place and maintain one copy of your task on each container instance.

Desired tasks

Specify the number of tasks to launch

4. Scroll down and click on **Load balancing** to expand the load balancing section:

An application load balancer and appropriate related resources have been pre-configured for you in this lab.

5. Enter and select the following to configure a public-facing load balancer for your Amazon ECS service:

Note: When you begin to select different options, other more specific options will appear in the form.

- **Load balancer type**: Select **Application Load Balancer**
- **Application Load Balancer**: Select **Use an existing load balancer**
- **Load balancer**: Select **ecslab-alb**
- **Listener**: Select **Use an existing listener**
- **Listener**: Select **80:HTTP**
- **Target group**: Select **Use an existing target group**
- **Target group name**: Select **ecslab-target-group**

Load balancing - *optional*

Load balancer type | Info

Configure a load balancer to distribute incoming traffic across the tasks running in your service.

Application Load Balancer

Load balancer

Select the load balancer you wish to use to distribute incoming traffic across the tasks running in your service.

ecslab-alb

Choose container to load balance

ecslab-blue-container 8081:8081

Listener | Info

Specify the port and protocol that the load balancer will listen for connection requests on.

Port	Protocol
80	HTTP

Target group | Info

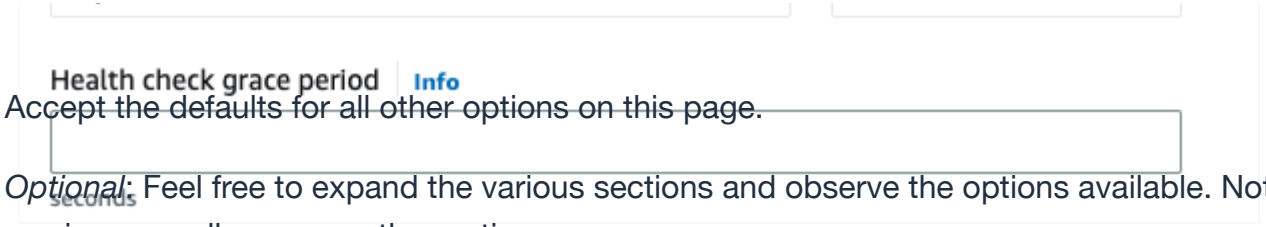
Specify whether to create a new target group or choose an existing one that the load balancer will use to route requests to the tasks in your service.

- ○ Create new target group
- ◉ Use an existing target group

Target group name

ecslab-target-group

Health check path	Health check protocol
/api/	HTTP



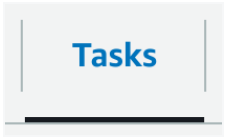
Accept the defaults for all other options on this page.

Optional: Feel free to expand the various sections and observe the options available. Notably, this page enables you to configure scaling for the service, as well as many other options.

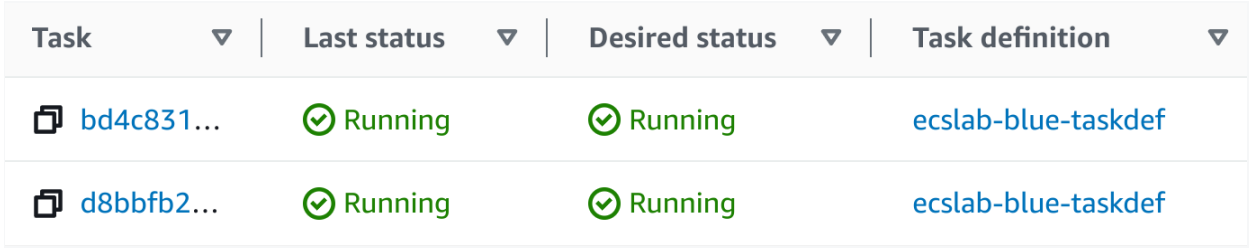
6. When ready, at the bottom of the page, click **Create** to finish creating your blue Amazon ECS service:

The **Cluster overview** page for your **ecslab-cluster** will load and you will see a blue notification at the top informing you the service is being created. After about a minute, you will see a green notification that it's been created successfully:

7. To view the tasks in your service, in the row of tabs, click **Tasks**:

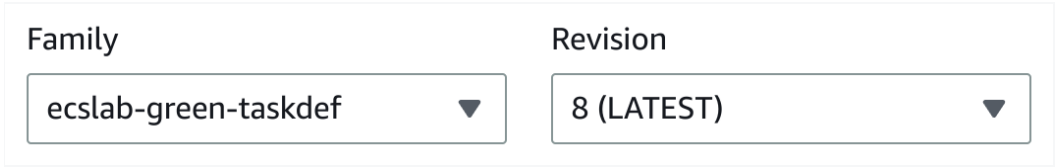


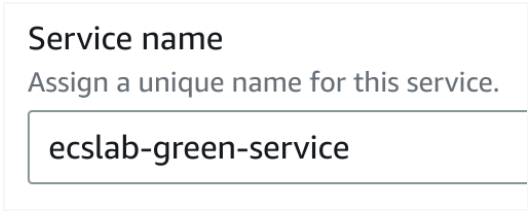
You will see two tasks reporting **Running** for **Last status**:

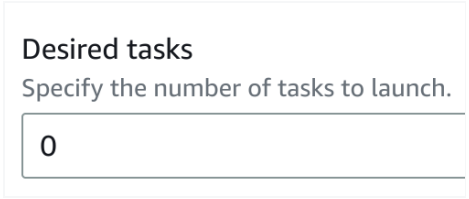


8. Repeat the instructions for creating a service, with the following changes:

- **Deployment configuration**:
  - **Family**: Select **ecslab-green-taskdef**
  - **Service name**: Enter *ecslab-green-service*
  - **Desired tasks***:* Replace **1** with *0*
- For all other options, configure the same values as you did for the blue service







Configure the **Load balancing** options to be the same as those for the blue service.

Notice that you have set **Desired tasks** to zero. A service with zero tasks will not launch any container instances upon creation. This allows you to prepare for future operations before your deployment is fully ready.

After clicking **Create** and waiting for your green service to deploy successfully, you will see two services with a status of **Active** on the **Services** tab of the **Cluster overview** page:

Notice that the blue service has two running tasks and the green service has zero.

Summary

In this lab step, you created two services for your blue and green applications. You learned how these services control the desired capacity. You started two tasks for the blue application upon service creation. These tasks launched and registered two container instances.