Introduction

Now that you know how the services interact and how to launch tasks into your Amazon ECS cluster, it is time to switch from one application to the next. A blue-green deployment is a technique that uses two identical production environments to reduce downtime. There are different cutover strategies, but generally, only one environment should be serving production traffic.

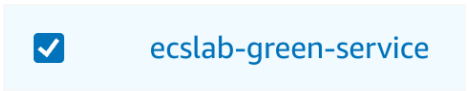
In this lab step, you will launch both applications with all of the container instances you need, including a brief overlap period, to demonstrate the load balancer's round-robin distribution and container resource efficiency.

Instructions

1. Navigate to the Clusters page of the Amazon ECS console.

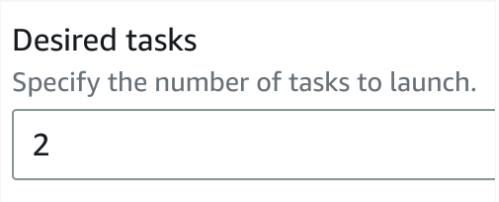
2. To access the cluster overview, click **ecslab-cluster**:

3. To select the green service, in the **Services** tab, click the checkbox next to **ecslab-green-service**:



4. To modify the green service, with it selected, click **Update**:

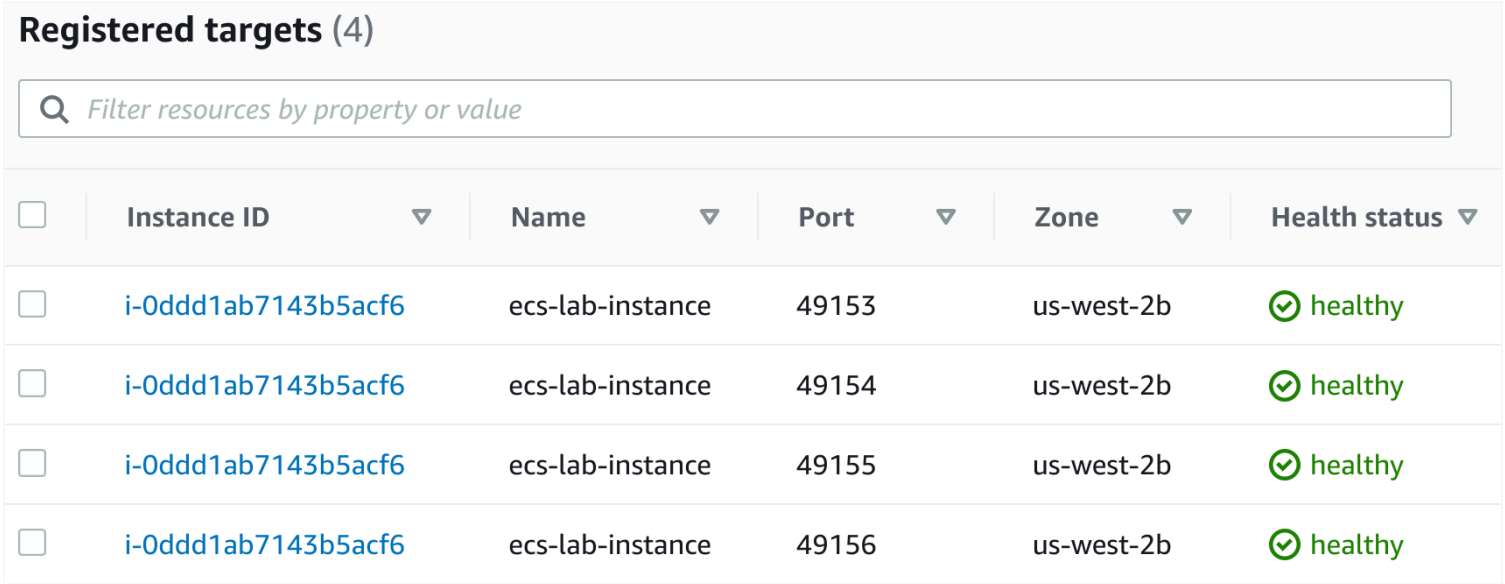
5. Change **Desired tasks** from **0** to *2*:



6. To make this change take effect, at the bottom of the page, click **Update**:

7. Return to the **Target Groups** section of the Amazon EC2 console and click **ecslab-target-group** to view details.

You will now see four instances under **Registered targets**:



The reduced resource requirements for a Docker container allow you to run multiple container instances on one EC2 instance. This works well for applications that require little resources, such as this simple message application.

8. Refresh your browser tab with the DNS name of the load balancer in the address bar.

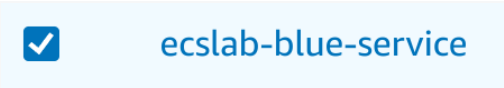
It may take several refreshes, but you will see the message change:

```
{"message":"Hello - I'm GREEN"}
```

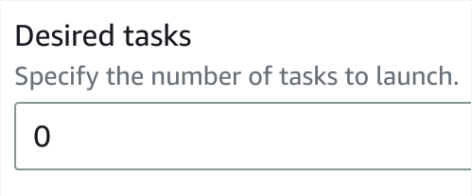
Right now you have both versions of the application running in their own pair of container instances.

9. Return to the cluster overview page for your cluster in the Amazon ECS console.

10. Select the **ecslab-blue-service** and click **Update**:



11. Reduce the **Desired tasks** field from **2** to *0*, and click **Update** at the bottom of the page:



Optional: Feel free to return to the target group page of the Amazon EC2 console and observe two of the registered targets being deregistered.

12. Refresh your browser tab with the DNS name of the load balancer in the address bar several times.

This time you will only see the message from the green application. It may take a couple of refreshes to see the change take effect.

By swapping the desired tasks of each service, you have manually replicated a blue/green deployment. In this lab, you used the Amazon ECS console for the purposes of learning. Be aware that in non-lab environments, this switching process is usually either partially, or fully automated.

When partially automated, a manual action may still be required. This is often referred to as a manual gate on the deployment and is used to ensure a person authorizes the switchover to the new deployment version.

When fully automated, sometimes automated testing is used to verify that the new deployment is functional before switching the deployment versions without manual intervention.

All of the services used in this lab (most prominently ECS, EC2, CodeBuild, and ECR) are well supported by the AWS command-line interface (CLI), AWS HTTP application programming interface (API), and AWS software development kits (SDK). Using these methods to create, configure, and operate the Amazon ECS and related services can result in fully automated deployments that are customized to your needs and workflow.

Summary

In this lab, you used AWS CodeBuild along with Amazon Elastic Container Registry to build and store docker images. You then created a new Amazon Elastic Container Service cluster, and the ECS task definitions and ECS services necessary to perform a blue/green deployment. Finally, you verified that the deployments were working, and manually switched from blue to green versions of the application.