## Introduction

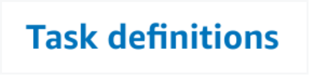
Task definitions are required to run Docker containers in Amazon ECS. They tell the services which Docker images to use for the container instances, what kind of resources to allocate, network specifics, and other details.

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

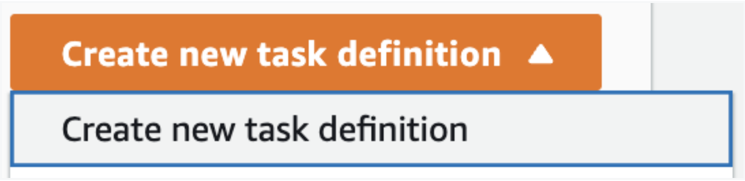
This is a somewhat simplified demonstration. In reality, you can define multiple containers and data volumes in a task definition, use variables, modify source files, or a number of other approaches to maintaining an application.

## Instructions

1. In the Amazon ECS console, in the left-hand menu, click **Task definitions**:



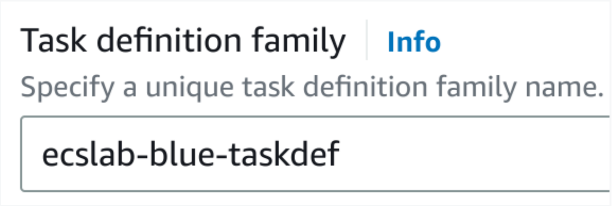
2. To begin creating a new task definition, in the top-right, click **Create new task definition**, and click **Create new task definition** in the dropdown that appears:



A multi-step form will load with Step 1 titled **Configure task definition and containers** will load.

3. To configure your task definition, enter and select the following, leaving all other options at their default:

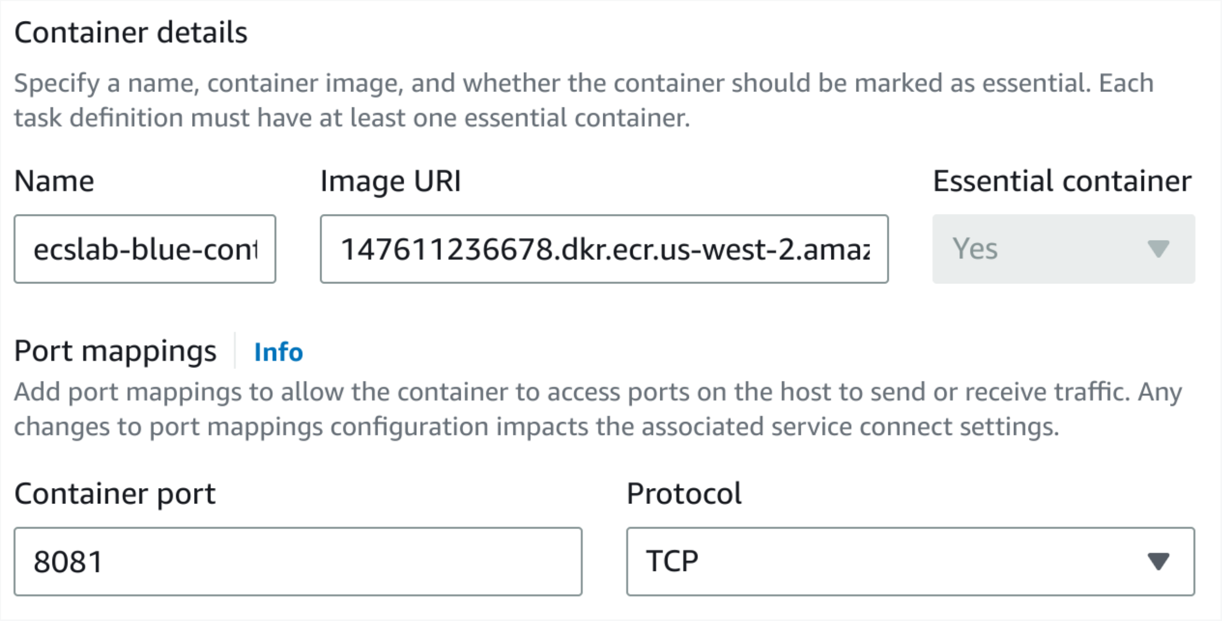
- **Task definition configuration**:
  - **Task definition family**: Enter *ecslab-blue-taskdef*
- **Container - 1**:
  - **Name**: Enter *ecslab-blue-container*
  - **Image URI**: Enter the URI of the **testblue** Docker image URI you made a note of previously
  - **Container port**: Change to *8081*



Task definition family Info

Specify a unique task definition family name.

ecslab-blue-taskdef



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 |
|---|---|---|
| ecslab-blue-cont | 147611236678.dkr.ecr.us-west-2.amaz | Yes ▼ |

Port mappings Info

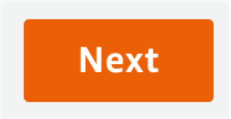
Add port mappings to allow the container to access ports on the host to send or receive traffic. Any changes to port mappings configuration impacts the associated service connect settings.

| Container port | Protocol |
|---|---|
| 8081 | TCP ▼ |

8081 is the port that the blue and green applications listen on when they are run.

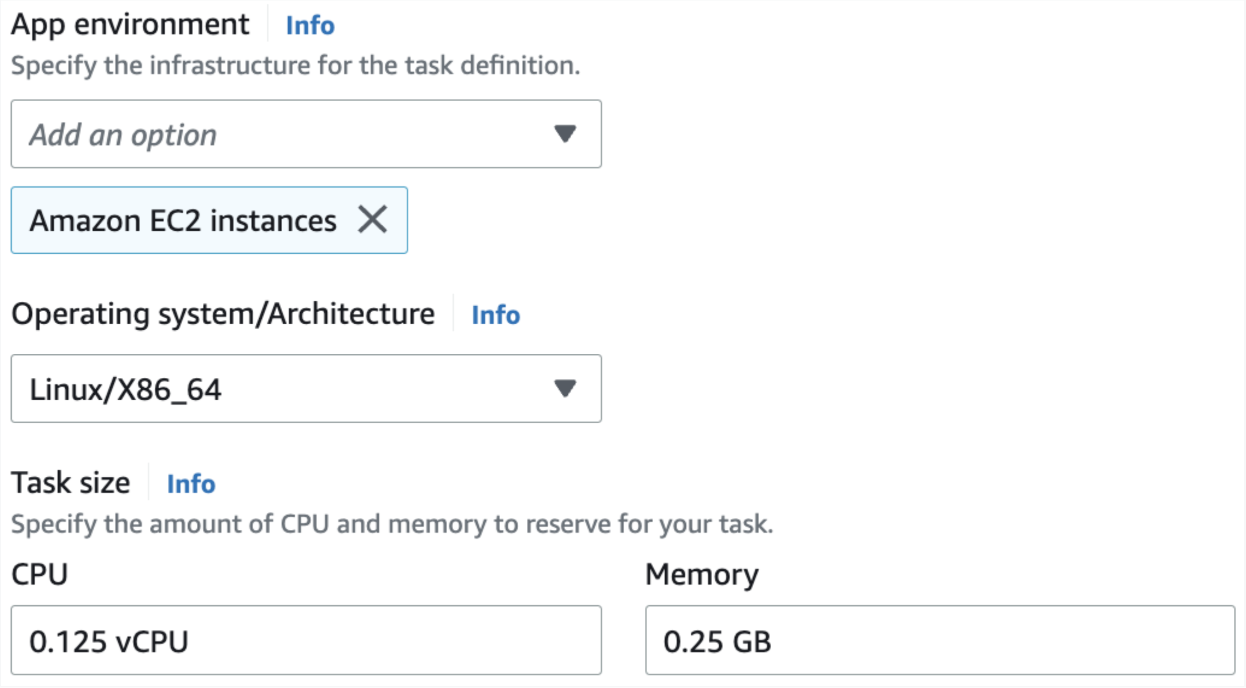
By default, the form selects **TCP** for the **Protocol**, and **HTTP**, for the **App protocol**. These values are appropriate for the blue and green applications.

4. To continue configuring your task definition, at the bottom of the page, click **Next**:



5. Enter and select the following:

- **Environment**:
  - **App environment**: Deselect **AWS Fargate (Serverless)**, and select **Amazon EC2 instances**
  - **CPU**: Enter *0.125 vCPU*
  - **Memory**: Enter *0.25 GB*



App environment

Specify the infrastructure for the task definition.

Amazon EC2 instances

Operating system/Architecture

Linux/X86_64

Task size

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

CPU: 0.125 vCPU

Memory: 0.25 GB

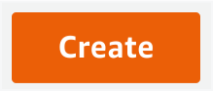
*Warning*: Setting task CPU and memory limits higher than the above may cause subsequent lab steps to not work correctly. The Amazon EC2 Auto Scaling group is configured to launch one t2.micro instance only.

Amazon ECS supports creating task definitions comprised of multiple containers. You have specified CPU and memory limits at the task level. You can also do so for each container in the task definition. In this lab, as there is only one container in the definition, you do not need to specify container-level limits.

6. To move to the next step of the form, scroll to the bottom and click **Next**.

The final page allows you to review the options you have selected.

7. To finish creating the task definition for the blue container, scroll to the bottom and click **Create**:



You will see an **Overview** page load for your task definition and a notification at the top of the page that it was created successfully:

Task definition successfully created

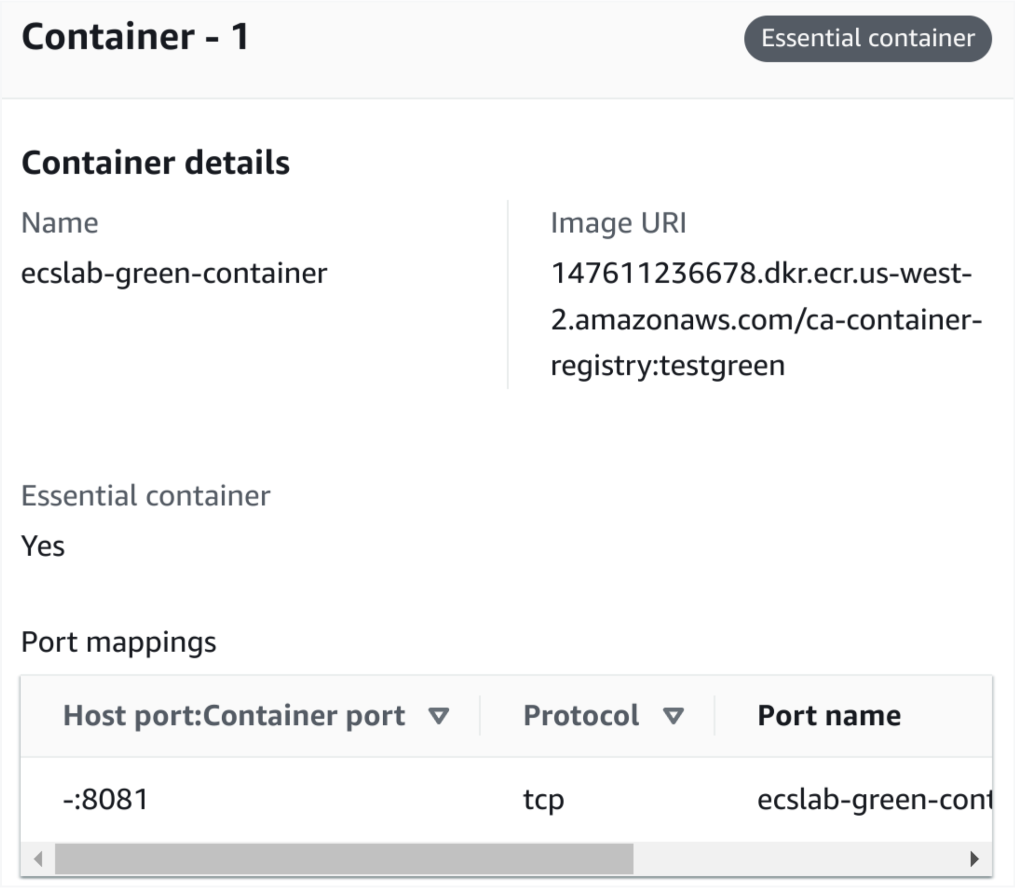
*Note*: You may see your task definition be assigned a **revision** number higher than one. Revisions are stored in an AWS account indefinitely even if they are marked inactive. If a student in a previous lab session created a definition with the same name, a new task definition will be assigned the revision of the previous definition, incremented by one.

8. Return to the task definitions page by clicking **Task definitions** in the breadcrumb navigation at the top of the page:

9. Repeat the instructions for creating a task definition, with the following changes:

- Enter *ecslab-green-taskdef* for the **Task definition family**
- Enter *ecslab-green-container* for the container name
- Use the URI of the **testgreen** image from your Amazon ECR repository for the **Image URI**
- For all other options, configure the same values as you did for the blue container

The review page of your green task definition creation form should look similar to:



Container - 1

Essential container

Container details

Name

ecslab-green-container

Image URI

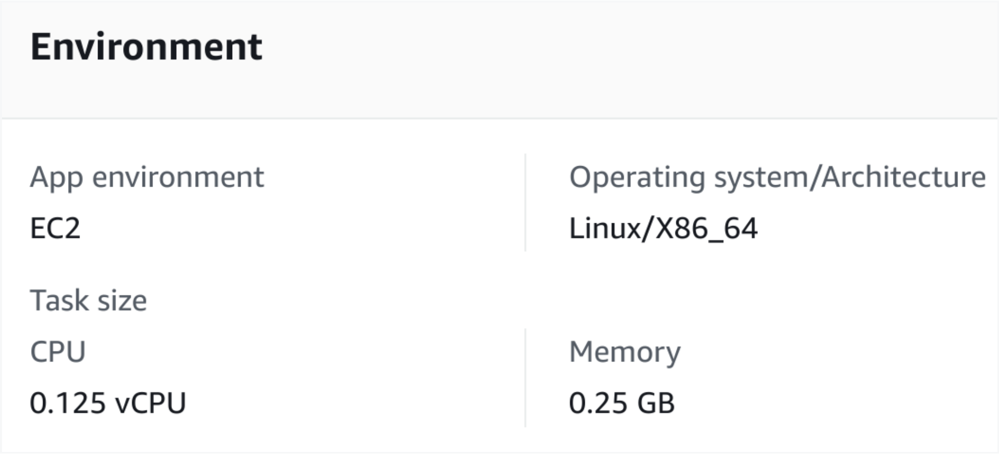
147611236678.dkr.ecr.us-west-2.amazonaws.com/ca-container-registry:testgreen

Essential container

Yes

Port mappings

| Host port:Container port ▽ | Protocol ▽ | Port name |
|---|---|---|
| -:8081 | tcp | ecslab-green-cont |



Environment

App environment

EC2

Operating system/Architecture

Linux/X86_64

Task size

CPU

0.125 vCPU

Memory

0.25 GB

## Summary

In this lab step, you created two task definitions for your application. You specified information about the Docker container images, resources, and networking details to use.

VALIDATION CHECKS

1 Checks

Check again ↻

✓ **Created an ECS Task Definition**

Check if an Amazon ECS task definition has been created

Amazon Elastic Container Service (ECS)