Note

The exercises in this course will have an associated charge in your AWS account. In this exercise, you create the following resources:

- AWS Identity and Access Management (IAM) policy and user (Policies and users are AWS account features, offered at no additional charge)
- AWS App Runner service
- Amazon Elastic Container Registry (Amazon ECR) repository

Familiarize yourself with **AWS App Runner pricing**, **Amazon ECR pricing**, and the **AWS Free Tier**.

Exercise 2: Using Amazon ECR and AWS App Runner

In this exercise, you create an Amazon ECR repository, authenticate to Amazon ECR, and then push your image to the Amazon ECR repository. You will then set up and deploy an AWS App Runner service.

Task 1: Creating an Amazon ECR repository

In this task, you will create an Amazon ECR repository.

1. In the AWS Management Console, choose **Services**, and then search for and open **Elastic Container Registry**.
2. If you have no existing repositories, choose **Get Started**. Otherwise, choose **Create repository**.
3. For **Repository name**, enter `first-container` and then choose **Create repository**.
4. Choose the **first-container** repository link and then choose **View push commands**.

   You should see the commands that you will use from your AWS Cloud9 instance.

Task 2: Authenticate to Amazon ECR

In this task, you will authenticate to Amazon ECR through the AWS Cloud9 IDE instance by running the push commands (which you saw in Amazon ECR).

1. In the AWS Management Console, choose **Services**, and then search for and open **Cloud9**.
2. Choose **Open IDE**.
3. In the AWS Cloud9 terminal, find the `ACCOUNT_ID` and the `REGION`, and insert these values into the `aws ecr` command.

```
export ACCOUNT_ID=$(aws sts get-caller-identity --output text --query Account)

export REGION="`wget -q -O - http://169.254.169.254/latest/meta-data/placement/re

aws ecr get-login-password | docker login --username AWS --password-stdin ${ACCOU
```

4. Add a new tag to `first-container` that references the Amazon ECR container registry.

```
docker tag first-container ${ACCOUNT_ID}.dkr.ecr.${REGION}.amazonaws.com/first-co
```

5. Push the image to Amazon ECR.

```
docker push ${ACCOUNT_ID}.dkr.ecr.${REGION}.amazonaws.com/first-container:latest
```

Task 3: Setting up AWS App Runner

In this task, you will set up and deploy an AWS App Runner service.

1. At the top left, choose the **AWS Cloud9** icon and then choose **Go To Your Dashboard**.
2. In the AWS Management Console, choose **Services**, and then search for and open **App Runner**.
3. Choose **Create an App Runner service**.
4. For **Container image URI**, choose **Browse**.
5. Choose **Image repository**, choose **first-container**, and then choose **Continue**.
6. If you haven't used Amazon ECR before, for **ECR access role**, choose **Create new service role**. Otherwise, you can keep **Use existing service role** selected.
7. Choose **Next**.
8. For **Service name**, enter `first-container` and choose **Next**.
9. Choose **Create & deploy**. *Wait for the service creation to complete.*
10. Choose the **Default domain** link.

    You should see the application running.

Challenge

You might recall that the containerized application takes `MESSAGE_COLOR` as an environment variable.

For this challenge, edit your service settings and add an environment variable in the expected color format (for example, `#0000ff`).

Wait for the service update to complete. Confirm the use of the environment variable by visiting the re-configured application.

Cleaning up

In this task, you delete some of the resources that you used for this exercise.

1. Delete the AWS App Runner service.
    - Open the **App Runner** dashboard.
    - Choose **first-container**.
    - Choose **Actions** and then choose **Delete**.
    - Confirm the deletion.
2. Delete the IAM role.
    - Open the **IAM** dashboard.
    - Choose **Roles**.
    - Delete **AppRunnerECRAccessRole** and confirm the deletion.