

Note

The exercises in this course will have an associated charge in your AWS account. In this exercise, you create or use 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 Cloud9 integrated development environment (IDE) instance
- Amazon DynamoDB table
- Amazon Elastic Container Service (Amazon ECS) cluster

Familiarize yourself with [AWS Cloud9 pricing](#), [Amazon Elastic Container Service pricing](#), and the [AWS Free Tier](#).

Exercise 4: Using Amazon ECS

In this exercise, you install AWS Copilot. You then use AWS Copilot to create the Amazon ECS cluster and its associated tasks. The backend Amazon DynamoDB table was created when you used the CloudFormation template in Exercise 1.

Task 1: Installing AWS Copilot and creating the backend infrastructure

In this task, you will install AWS Copilot. You will also create the backend infrastructure.

1. In the AWS Management Console, open AWS Cloud9.
2. Open your AWS Cloud9 IDE instance.
3. Install AWS Copilot.

```
curl -Lo /tmp/copilot https://github.com/aws/copilot-cli/releases/latest/download/copilot

# copilot also needs the region configured
export REGION=$(curl -s -o /dev/null -w '%{url}' http://169.254.169.254/latest/meta-data/placement/region)

aws configure set region ${REGION}
```

4. Deploy the service and create the backend infrastructure.

NOTE: This process can take 5–10 minutes to complete.

```
copilot init \  
--app corpdirectory \  
--type "Backend Service" \  
--name "service" \  
--dockerfile ./directory-service/Dockerfile \  
--deploy
```

5. Observe that the backend has an endpoint (service discovery endpoint).

This endpoint is outputted after you run the previous command, and it should look like the following example:

```
service.test.corpdirectory.local:80
```

Task 2: Viewing the created resources

In this task, you will view the various backend resources.

1. View the resources that AWS Copilot created.

```
copilot app ls  
  
copilot app show corpdirectory  
  
copilot env ls  
  
copilot env show --name test  
  
copilot svc ls  
  
copilot svc show --name service  
  
copilot svc status --name service  
  
copilot svc logs --name service
```

Task 3: Deploying the frontend

In this task, you will exec into the container. You will then view the various resources that were created. Finally, you will prepare and deploy the frontend.

1. Install the AWS Session Manager plugin and exec into the container.

```
copilot svc exec --name service  
  
(y/N)y
```

2. From inside the container, check the Domain Name System (DNS) entry from AWS Cloud Map.

```
dig A service.test.corpdirectory.local

curl service.test.corpdirectory.local/employee

exit
```

Next, you will view the AWS Cloud Map resources that were created through the integrations with Amazon ECS.

3. View the DNS_PRIVATE namespace named *test.corpdirectory.local*. `bash aws servicediscovery list-namespaces`
4. View the service named *service*. `bash aws servicediscovery list-services`
5. From the output of the previous command, copy the service ID.
6. View the instance (replace the `<service-id>` placeholder with the ID that you copied in the previous step).

```
aws servicediscovery list-instances --service-id <service_id>
```

7. Prepare the frontend.

```
copilot svc init \
--app corpdirectory \
--svc-type "Load Balanced Web Service" \
--name "frontend" \
--dockerfile ./directory-frontend/Dockerfile
```

8. Update the newly created `copilot/frontend/manifest.yml` file with the directory service.

NOTE: This process can take 5–10 minutes to complete.

```
cat << EOF >> copilot/frontend/manifest.yml

variables:
  API_ENDPOINT: http://service.test.corpdirectory.local:80
EOF
```

9. Deploy the frontend.

```
copilot svc deploy --name frontend --env test
```

10. Visit the application, which should look like the following example.

You can access your service at `http://corpd-Publi-5ACV8N2M8EBW-012345678912.us-east-`

Challenge

Inspect `copilot/service/manifest.yml` and `copilot/frontend/manifest.yml`.

For this challenge, increase the *number of tasks* that host the services, and redeploy the application.

Cleaning up

1. Delete the AWS Copilot resources and confirm the deletion.

```
copilot app delete
```

```
(Y/n)y
```

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections, feedback, or other questions? Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>. All trademarks are the property of their owners.