**AWS Administration**

**Lab Title**: Breaking the Monolith

**Lab Author:** AWS Academy

**Version:** 3.0 **Points: 20**

**Purpose**: A guided lab on containerizing a monolithic application into a Docker container running on AWS’s Elastic Container Service and then breaking it into Microservices.

**What You’ll Need:** Access to the AWS Academy Architecture Course (see instructor) Note this lab CAN take up to three hours to complete.

**Instructions:**

1. Go to the AWS Academy: <https://www.awsacademy.com/LMS_Login>
2. Instructions for this lab are in the AWS Academy course: AWS Academy Cloud Architecting - Module 13

Lab Title:



1. Your instructor enrolled you in that course so you would access to this lab - if you do not see that course in the AWS Academy Canvas learning website see your instructor for access.
2. Follow all lab instructions in the lab itself EXCEPT where my updates are noted below - this is a LONG and COMPLICATED lab and you may need to go back and troubleshoot steps to get the services working correctly.

**\*\*\* READ BEFORE STARTING \*\*\* Notes about the lab / Corrections / Changes**

1. When building the load balancer the lab instructions reference the old interface steps - you \*may\* be building in the new interface so you’ll need to adjust your configuration accordingly. The big difference is when creating the target group:
   1. When you Build Your Target Group (Step 81)

Graphical user interface, text, application, email

Description automatically generated

* 1. When constructing the Target Group **DO NOT** register those existing EC2 instances! You’ll be registering your ECS Service as the target for the load balancer in a later step - the instructions point this out but you may miss it and it will affect the entire operation of the load balancer and hence the lab.
  2. After building the target group go back to the OPEN BROWSER TAB that is STILL RUNNING the load balancer creation wizard and refresh the target group selection and choose the target group you JUST created:

Graphical user interface, text, application, chat or text message

Description automatically generated

1. In Step #101 you MUST modify the Service IAM Role to select the existing role: AWSServiceRoleforECS otherwise the ECS Service you are creating will **NOT** launch!

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Clarification - in steps 108 through 110 you are going to the Load Balancer itself in the EC2 Service area and locating the DNS entry:

Graphical user interface, text, application, email

Description automatically generated

1. In Task 6.2: Configuring the Application Load Balancer - similar to before when you build these THREE target groups do NOT include the EC2 instances - just create the groups with no instances selected as these will eventually be routing requests to the three ECS services. When you get done building all three target groups they should look like this:

A screenshot of a computer

Description automatically generated with medium confidence

1. Step #223 - Make sure you selecting to edit the rules from this hyperlink:

Graphical user interface, text, application, email

Description automatically generated

1. Pay particular attention at step #240 and MAKE SURE your load balancer listener rules are exactly like the ones shown in the screenshot!

**REQUIRED SCREENSHOTS – GET THESE BEFORE ENDING THE LAB!**

**Screenshot of the 4 task definitions that you built in “ECS | Task Definitions”**

**Graphical user interface, application

Description automatically generated**

**Screenshot of the 4 ECS services that you built in “ECS | Clusters | Your Cluster | Services Tab” mb-ecs-services should show 0 running tasks**

**Graphical user interface, application, table

Description automatically generated**

**Screenshot of the target groups that you built in “EC2 | Load Balancing | Target Groups**

**Graphical user interface, application, table

Description automatically generated**

**Screenshot of your “users” microservice responding to an api call in the browser**

[**http://dns-name-of-your-load-balancer/api/users**](http://dns-name-of-your-load-balancer/api/users)

**Graphical user interface, text, application, email

Description automatically generated**

**“SUBMIT” your lab to the auto grading feature - wait a minute and these choose “Grades” - Put a screenshot of your Total Score x/80 here**

**Graphical user interface, text, application

Description automatically generated**

Graphical user interface, text, application

Description automatically generated

**If you don’t have an 80/80 score you can go back and look for the issues if you desire - however note this:**

If the api call above ([**http://dns-name-of-your-load-balancer/api/users**](http://dns-name-of-your-load-balancer/api/users)) responds with the users from the database AND your screenshot of the ECS Services shows 0 running tasks of the mb-ecs-service (the original one you setup that is not needed as your three micro services and processing the api requests) your ECS cluster is functioning correctly and I will consider it as 100% complete.