CSC4160 作业 3：使用 AWS Lambda 进行模型服务和冷启动性能分析（6 分）

截止日期：2024 年 10 月 24 日，23:59

＃＃＃ 姓名：

学生证号：

＃＃ 概述

在此作业中，您将学习使用 AWS Lambda 和 API Gateway 将机器学习模型部署为无服务器应用程序。您将创建 Docker 映像，将其推送到 Amazon Elastic Container Registry (ECR)，并对已部署的应用程序进行负载测试。此外，您还将分析与无服务器功能相关的冷启动现象。

我们将使用著名的 IRIS 数据集，以保持机器学习模型的简单性并专注于无服务器部署过程。该数据集包括萼片长度、萼片宽度、花瓣长度和花瓣宽度四个特征，并将样本分为三类：山鸢尾、变色鸢尾和维吉尼亚鸢尾。

＃＃＃ 成分

1. **Lambda 函数开发**

- 实现“lambda_handler”函数。

2. **环境设置**

- 设置您的本地开发环境。

3. **Docker 镜像创建**

- 制作一个 Docker 映像，它将使用经过训练的模型生成预测。

4. **ECR 存储库设置**

- 创建 AWS ECR 存储库并将您的 Docker 映像推送到 AWS ECR。

5. **在 AWS 控制台中创建 Lambda 函数**

- 使用容器映像创建 Lambda 函数。

6. **API 网关配置**

- 使用 API 网关访问预测 API

7. **负载测试与分析**

- 使用 Locust 对已部署的 API 执行负载测试。
- 绘制结果来观察冷启动趋势。
- 分析冷启动和热请求响应时间之间的差异。

指示

1. Lambda 函数开发

您将获得“预测”函数和模型文件；您的任务是实现“lambda_handler”函数。

lambda_handler 函数执行以下任务：

- 提取“值”：它从传入事件中检索输入的值，这些值是用于进行预测的特征。
- 调用预测函数：它调用预测函数，传递提取的值以根据机器学习模型生成预测。
- 返回预测结果：最后将预测结果格式化为JSON响应并返回给调用者。

<详细信息>

实现 `lambda_handler` 的步骤>

从事件中提取输入：

- 您将在事件的“主体”内收到输入的功能。
- 将此“body”解析为 JSON 并检索“values”。
- 您还可以处理任何可能的错误，例如缺少输入或无效的 JSON。

调用“预测”函数：

- 提取“值”后，将它们传递给“预测”函数，该函数将返回预测列表。

格式化并返回响应：

- 以 JSON 响应形式返回预测。 </详情>

<详细信息>

测试函数

使用模拟输入进行测试：

您可以通过 AWS Lambda 控制台模拟对“lambda_handler”的输入。例如，事件可能如下所示：

```
{
“主体”: “{\“值\”: [[5.1, 3.5, 1.4, 0.2]]}”
}
```

模拟预测：

如果不想上传模型就想进行测试，可以暂时模拟预测函数返回一个模拟结果。

在 AWS Lambda 中测试：

使用 AWS Lambda 控制台通过示例事件测试您的函数，或者您可以设置 API 网关并从那里发送请求。

</详情>

2. 环境设置

在你的机器上设置本地开发环境：

- 为您的操作系统安装 Docker Desktop：https://www.docker.com/
- 安装 AWS CLI：https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
- 确保已安装 Python 3 和 pip。
- （可选但推荐）安装 Git：https://git-scm.com/downloads
- 配置您的AWS凭证：

<详细信息>

AWS 凭证配置

要配置您的 AWS 凭证，请按照以下步骤操作：

1. **访问您的 AWS 凭证**：在 Vocareum 主页上，导航至“帐户详细信息”，然后导航至“AWS CLI”。复制提供的访问密钥 ID、秘密访问密钥和会话令牌。

2. **创建或打开凭证文件**：找到您的 AWS 凭证文件：

-**macOS**：`~/.aws/credentials` -**Windows**：`C:\Users\%UserProfile%\.aws\credentials`

如果该文件不存在，请使用纯文本编辑器创建它。

3. **添加您的凭证**：使用以下格式将访问密钥 ID、秘密访问密钥和会话令牌粘贴到文件中。添加“region”行（您可以使用任何区域，例如“us-east-1”）：

```
[默认]
region=us-east-1 # 添加此行。
aws_access_key_id=您的访问密钥 ID
aws_secret_access_key=你的秘密访问密钥
aws_session_token=你的会话令牌
```

将“YOUR_ACCESS_KEY_ID”、“YOUR_SECRET_ACCESS_KEY”和“YOUR_SESSION_TOKEN”替换为您从 Vocareum 复制的值。

4. **保存文件**：确保文件已保存，并且只有您才有权访问它。

5. **重要安全注意事项**：切勿共享您的 AWS 凭证。将它们视为密码。不要将此文件提交给版本控制（例如 Git）。将 `.aws/credentials` 添加到您的 `.gitignore` 文件中。考虑在生产环境中使用更安全的方法来管理凭证。

</详情>

3. Docker 镜像创建

在您的本地机器上：

- 使用提供的Dockerfile创建Docker镜像：

```
docker build-t iris_image 。
```

- 本地运行 Docker 容器：

```
docker run -it --rm -p 8080:8080 iris_image:latest
```

这里我们映射端口8080。

- 通过执行"test.py"来验证图像是否正常运行。

4. ECR 存储库设置

开始之前，请登录 AWS 管理控制台。然后，在本地计算机上执行以下步骤。

- 创建 ECR 存储库：

```
aws ecr 创建存储库 --存储库名称 iris-registry
```

- 使用 ECR 验证您的 Docker 客户端：

```
aws ecr get-login-password --region us-east-1 | docker login --用户名 AWS --
password-stdin <aws_account_id>.dkr.ecr.us-east-1.amazonaws.com
```

- 获取图像ID：

```
docker 镜像
```

- 标记并推送您的 Docker 映像：

```
docker 标签 <image_id> <aws_account_id>.dkr.ecr.us-east-1.amazonaws.com/iris-
registry:latest

docker push <aws_account_id>.dkr.ecr.us-east-1.amazonaws.com/iris-registry:latest
```

5. Lambda 函数创建

- 在 AWS 控制台中，使用您构建的现有容器映像创建 Lambda 函数，并选择"LabRole"作为执行角色。

6. API 网关配置

- 通过 AWS 控制台使用 API 网关为您的 Lambda 函数创建 REST API。
- 使用“curl”（Linux）在本地机器上测试你的 API：

```
curl --header "Content-Type: application/json" --request POST --data "{\"values\":
[[<value1>, <value2>, <value3>, <value4>]]}" https://<your_api_id>.execute-api.
<region>.amazonaws.com/default/<your_lambda_function>
```

或使用 `Invoke-WebRequest` (Windows)：

```
Invoke-WebRequest-Method Post-Uri“https: //<your_api_id>.execute-api。
<region>.amazonaws.com/default/<your_lambda_function>”`
-Headers @{ “Content-Type” = “application/json” } `
-Body '{"值":[[<值1>, <值2>, <值3>, <值4>]]}'
```

7. 负载测试和分析

负载测试

在您的本地机器上，使用提供的 Locust 负载测试脚本来评估已部署 API 的性能。

- 安装 Locust

```
pip 安装 locust
```

- 使用以下方法运行 Locust 测试：

```
locust -f locustfile.py --host https://<your_api_gateway_id>.execute-api.us-east-
1.amazonaws.com --users 100 --spawn-rate 20 --run-time 60s --
csv“locust_logs/test”--csv-full-history --
html“locust_logs/test_locust_report.html”--
logfile“locust_logs/test_locust_logs.txt”--headless
```

对于 Windows 用户，设置“locust”的 PATH，或者直接使用“locust.exe”，指定其路径，例如：

```
c:\users\用户\appdata\roaming\python\python39\scripts\locust.exe
```

分析

使用 Google Colab 上的性能分析笔记本分析结果。上传您的日志并运行笔记本“performance_analysis.ipynb”。在绘制直方图之前，请填写估计的冷启动时间（在“”中），以比较冷启动和热请求期间的响应时间。

如果您在 `.ipynb` 中包含所需的图表，您将获得 1 分：折线图、冷启动直方图和热请求直方图。此外，如果您清晰地解释您的分析，您将获得 0.5 分。

＃＃ 问题

了解 AWS Lambda、API 网关和 ECR

1. **AWS Lambda 函数**（0.5 分）：

Lambda 函数在无服务器部署中的作用是什么？`lambda_handler` 函数如何处理请求？

答案：<你的答案在这里>。

2. **API 网关与 Lambda 集成** (0.5 分)：

解释一下此部署过程中 API Gateway 的用途。它如何将请求路由到 Lambda 函数？

答案：<你的答案在这里>。

3. **ECR 角色** (0.5 分)：

ECR 在此部署中扮演什么角色？它如何与 Lambda 集成以管理容器化应用程序？

答案：<你的答案在这里>。

冷启动现象分析

4. **冷启动与热请求** (1 分)：

提供您的分析，比较冷启动期间请求与热请求的性能（基于您在“performance_analysis.ipynb”中获得的折线图和直方图）。讨论响应时间的差异以及在负载测试期间观察到的任何值得注意的模式。

答案：<你的答案在这里>。

5. **含义和策略**（0.5 分）：

讨论冷启动对无服务器应用程序的影响及其对性能的影响。可以采用哪些策略来减轻这些影响？

答案：<你的答案在这里>。

提交要求

请通过 BlackBoard 以一个 `.zip` 文件的形式提交您的作业，其中包括以下文件：

- `README.md`（包含所有问题的答案）（3 分）
- `lambda_function.py`（您对 Lambda 函数的实现）（1 分）
- 提取“值”
- 调用预测函数
- 返回预测结果
- 为您的某个 Lambda 函数调用提供 CloudWatch 日志事件。（0.5 分）
- 使用 `curl` 或 `Invoke-WebRequest` 成功请求和响应的屏幕截图，显示正确的预测输出。（0.5 分） - `performance_analysis.ipynb`，包括：
- 折线图、冷启动直方图和热请求直方图的图形。（0.5 分）

- `test_locust_logs.txt`（Locust 负载测试日志）（0.5 分）

作者

此作业由 Juan Albert Wibowo 于 2024 年 10 月 8 日为 CUHKSZ 2024 年秋季 CSC4160：云*计*算 创建。