Stable Diffusion API 架构

待补充

环境准备

1. 一台 EC2，有 root 权限，普通 m5/m6i large 即可
2. 下载 `stable-diffusion-webui-api-main` 代码 repo

```
git clone git@github.com:aws-samples/amazon-ai-with-slack-bot.git
```

```
cd amazon-ai-with-slack-bot/stable-diffusion-webui-api-main
```

构建容器镜像

执行脚本，执行前先阅读下方内容后，再进行

```
cd stable-diffusion-webui-api-main # 进入 repo 目录
build_and_push.sh.lite us-east-1 # us-east-1 为 region，可以改为自己的 region
```

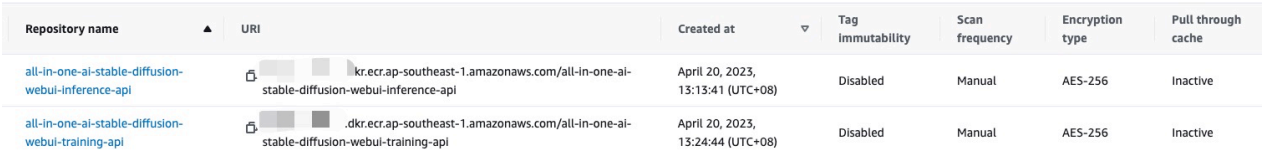
这个脚本会做以下操作：

1. 在本地 build 两个容器镜像，分别用于推理和训练，**这里只介绍推理部分原理**
    1. 这两个镜像可以自定义名称，修改脚本 inference_image, training_image 变量即可
    2. 这两个镜像的 image 代码几乎全部是开源 stable-diffusion-webui 代码，在 stable-diffusion-webui-api-main/stable-diffusion-webui 目录下
    3. 可以参考 docker build 时候的 Dockerfile：Dockerfile.inference.lite, Dockerfile.training.lite
2. 把本地 build 的容器镜像 push 到 ECR（如不存在相应的 ECR repo，则会创建一个新的）

3. 如果 build training 镜像报错 `fatal: destination path '/opt/ml/code/extensions/sd_dreambooth_extension' already exists and is not an empty directory.`，则修改一下 Dockerfile.training.lite 文件

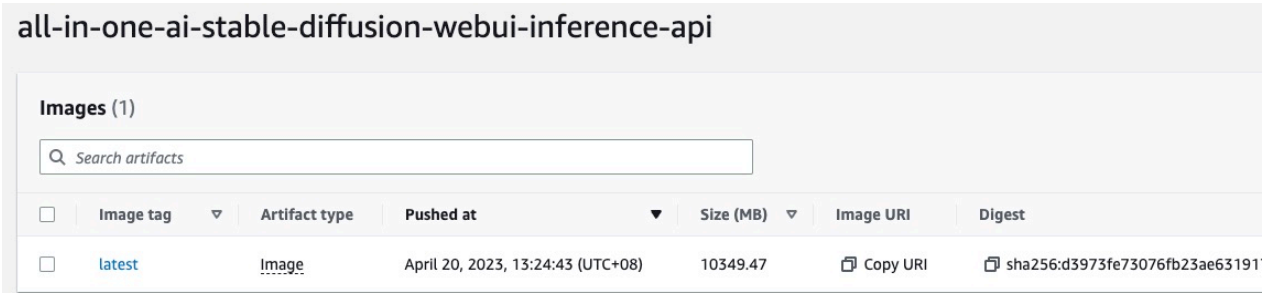
```
FROM public.ecr.aws/l1s7l7p8/all-in-one-ai-stable-diffusion-webui-training:latest

COPY train.py /opt/ml/code
COPY stable-diffusion-webui /opt/ml/code/

RUN rm -rf /opt/ml/code/extensions/sd_dreambooth_extension # 添加这一行，先删掉以前的文件
RUN git clone https://github.com/xieyongliang/sd_dreambooth_extension.git /opt/ml/code/extensions/sd_dreambooth_extension
```

4. 脚本跑完后，可以去 ECR 检查一下镜像是否 push 成功
    1. ECR repo 成功创建



Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type	Pull through cache
all-in-one-ai-stable-diffusion-webui-inference-api	kr.ecr.ap-southeast-1.amazonaws.com/all-in-one-ai-stable-diffusion-webui-inference-api	April 20, 2023, 13:13:41 (UTC+08)	Disabled	Manual	AES-256	Inactive
all-in-one-ai-stable-diffusion-webui-training-api	.dkr.ecr.ap-southeast-1.amazonaws.com/all-in-one-ai-stable-diffusion-webui-training-api	April 20, 2023, 13:24:44 (UTC+08)	Disabled	Manual	AES-256	Inactive

    2. ECR repo 里有镜像



5. 这里的镜像在未来会在创建 Sagemaker Inference Endpoint 时用上

打包模型上传

这里的模型可以是自己训练的模型，也可以是网上（huggingface, civitai 等）上下载的模型，目前支持 `.ckpt` 和 `.safetensors` 格式

1. 创建三个空目录

```
mkdir Stable-diffusion
mkdir Lora
mkdir ControlNet
```

2. 这三个目录的作用和开源 stable-diffusion-webui 一致，大小写敏感

    3. 如果目录结构有问题，则在后面创建 Sagemaker Inference Endpoint 是会创建失败
2. 在三个目录里分别放入你自己的（或者从网上下载的）文件
    1. `.ckpt` / `.safetensors` 格式的模型放到 Stable-diffusion 目录下
    2. lora 的文件放到 Lora 目录下
    3. 这里的操作和本地自建单机版 stable-diffusion-webui 添加模型和 lora 是一样的
3. 打包成 tarball 并传到 s3

```
tar zcvf model.tar.gz ControlNet/ Lora/ Stable-diffusion/
aws s3 cp model.tar.gz s3://{bucket}/model.tar.gz # 传到 S3 任意位置都可以，在后面 Sagemaker Inference Endpoint 部署时指定即可
```

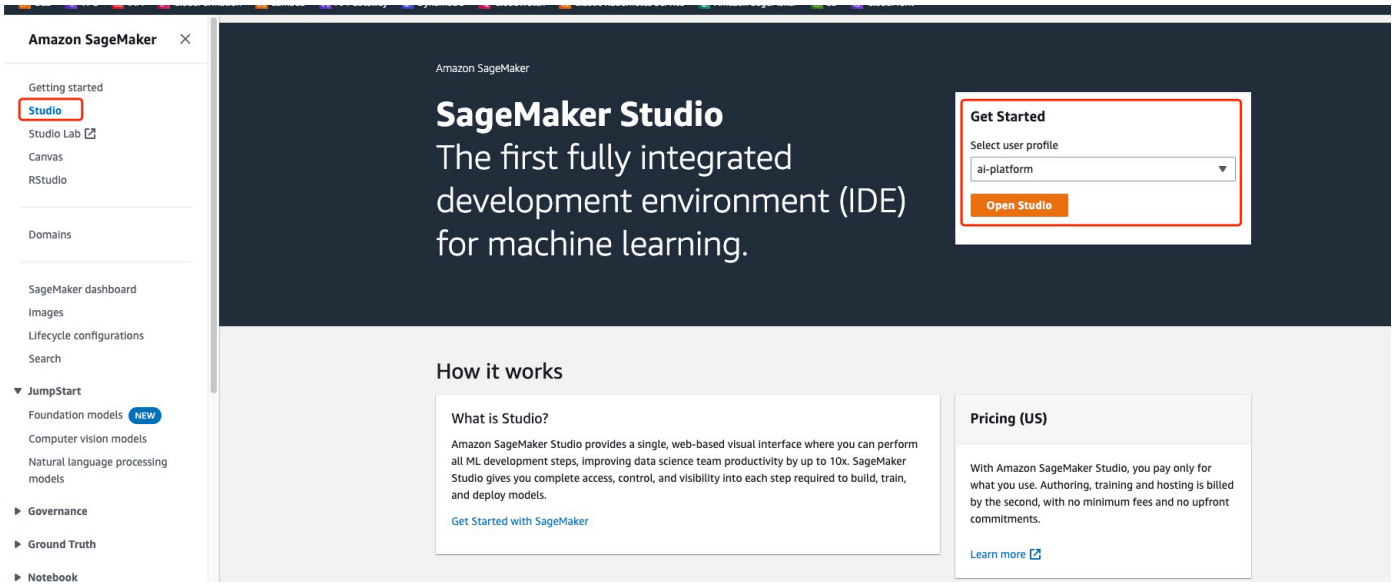
创建 Sagemaker Inference Endpoint

推荐使用 Sagemaker Studio Notebook 来创建 Inference Endpoint，更容易调试和修改

目前 Notebook 可以简单理解为：

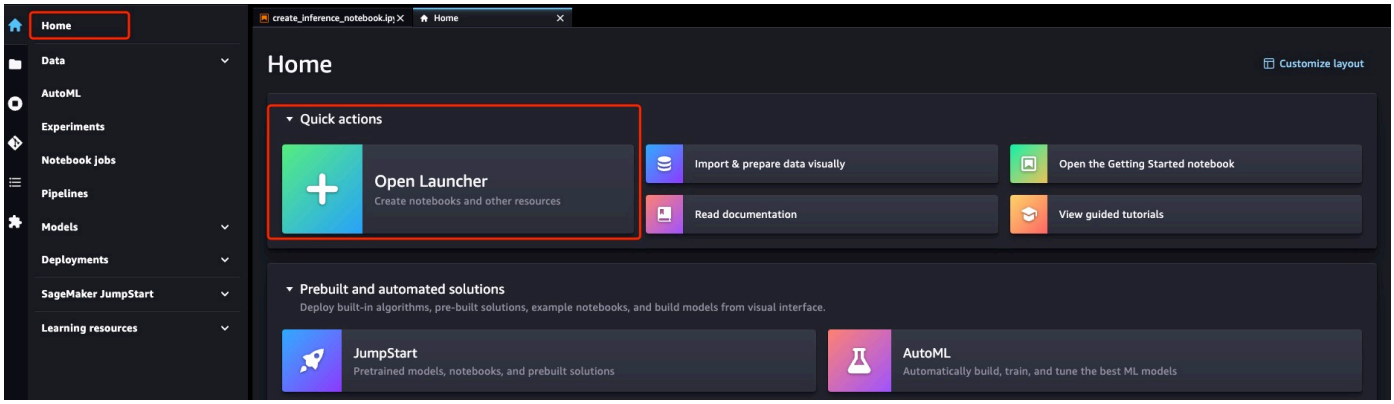
1. 你可以理解为就是一台机器，有机型和大小的概念（c5.large / c5.xlarge / g5.large ...）
2. 专门用于执行机器学习任务的环境，里面预置了很多依赖和软件库（Python）
3. 可以单步执行 Python 和 Shell 代码
4. 可以写注释

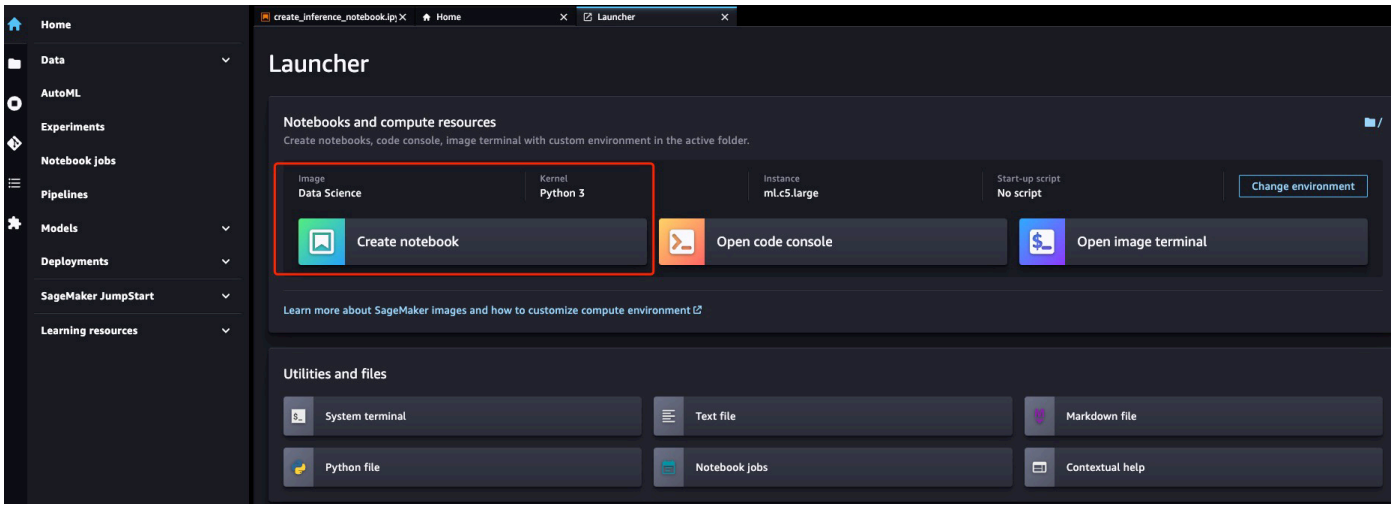
在 Sagemaker Studio 中打开 `create_inference_endpoint.ipynb` 这个 Notebook：



如果这里尚无 Studio，则初始化一个 Domain 和 User profile 即可

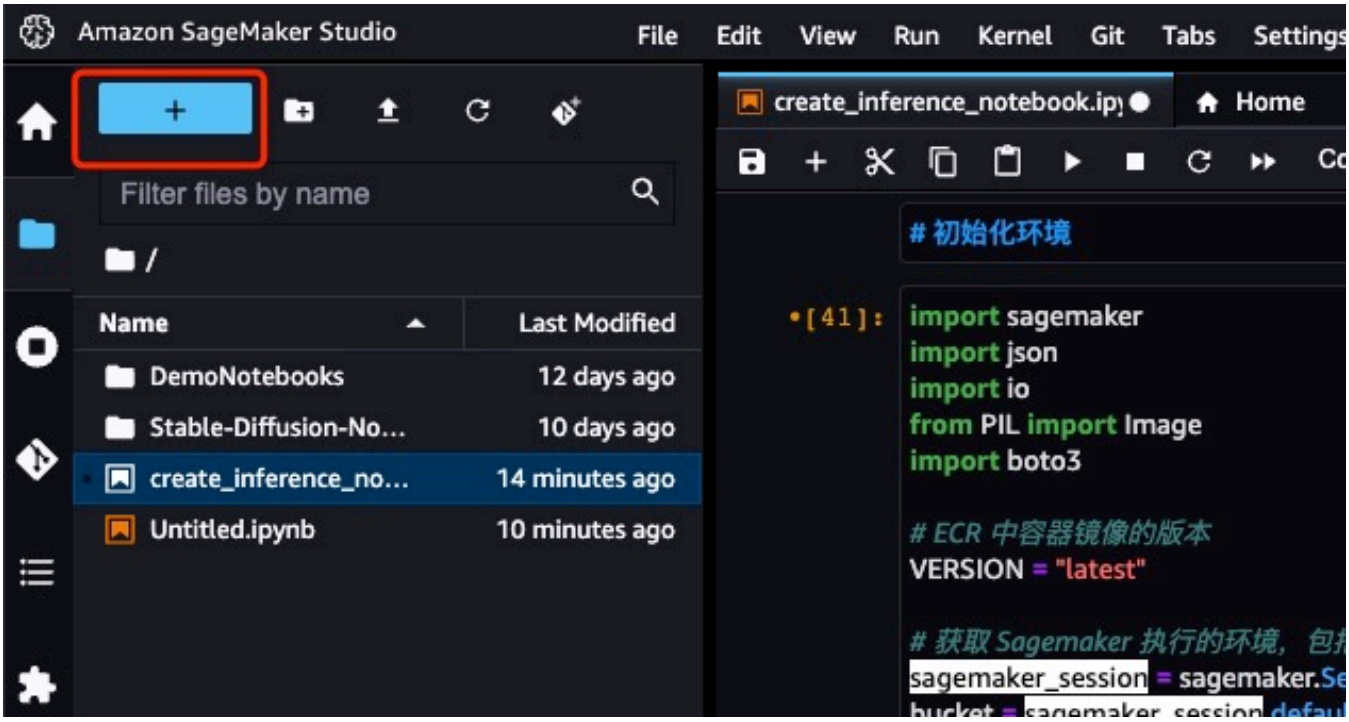
创建 Notebook





机型选择 c5.large 即可，其他默认

将 `create_inference_endpoint.ipynb` 文件上传到 Notebook



打开`create_inference_endpoint.ipynb`这个 Notebook，剩余步骤和解释请参考 Notebook