Sagemaker Async Inference on Video files.

There are two parts to this implementation.

1. Sagemaker Inference Endpoint Creattion
2. Https Endpoint with Authorization and custom input/output

1. Sagemaker Inference Endpoint Creattion

This part can be deployed by executing all cells in the sagemaker.ipynb file. The instructions for each cell and why we need this step are mentioned in the notebook file. Also some test script to invoke the sagemaker endpoint directly has been written and included in the notebook itself as well for local testing.

2. Https Endpoint with Authorization and custom input/output

After the successful sagemaker endpoint creation. We can go to the `infra` directory using `cd infra` command and deploy the infrastructure using following command:

```
sam sync
```

Details

Prerequisites

To deploy the https endpoint for sagemaker inference, you will need to have AWS SAM CLI and python installed. If you haven't installed it yet, follow the instructions below.

Installing Dependencies

- Install AWS SAM CLI: To install AWS SAM CLI please refer to the official AWS SAM CLI documentation here
- Install Python

Deployment Via Sam

Follow these steps to deploy the entire api endpoint alongside associated resources:

1. Navigate to the project directory:
2. Execute the following commands and pass the desired password as a parameter:

```
cd scripts
.\store_api_pass mysupersecurepass
```

Note: You must pass this password in your headers to the api call with header name: `Auth-Pass`

3. Open the `template.yaml` file and update the following Parameter in as per your requirements, I have specified the default values but in case we want to change.
   - `OutputVideoBucketName` : The bucket where output videos are stored. The default bucket name is `sm-ball-tracking-output-blobs` .
   - `SageMakerEndpointName` : The Name of Sagemaker Endpoint (you specify in the sagemaker.ipynb file to be deployed). The default bucket name is `ball-tracking-v7` .
   - `InputBucketName` : The bucket where the input request payload(json) would be stored. Please not that this is not the bucket for input. The default bucket name is `sm-ball-tracking-inputs` .
   - `OutputLabelsBucketName` : The bucket where output labels for inference are stored. The default bucket name is `sm-ball-tracking-output-labels` .
4. The default region of deployment is set to be default region of your aws cli profile. If you want to deploy in some other region add `region = "us-east-1"` in samconfig.toml. Please change that as required.
5. Build and deploy the application using the following command:

```
sam sync
```

6. Go to Api Gateway console. Click on `Api-->AwsApiGateway-->Api Keys-->Api`
   - Click on show key and note it down. You must pass this key as `x-api-key header

Manually Update Api-Pass and x-api-key

In order to Manually update the `Api-Pass` and `x-api-key header` values, Follow the following instructions

For Api-Pass

- Sign in to your AWS account and open the AWS Management Console.

- Ensure that the region is set to “ca-central-1”.
- Navigate to the "Systems Manager" service.
- Click on "Parameter Store" in the left-hand menu.
- Use the search bar to find the API_AUTH_PASSWORD parameter.
- Click on the parameter name to open its details.
- Click the "Edit" button on the top right corner.
- Enter the new password securely in the "Value" field.
- Optionally, add a description if needed.
- Click "Save changes" to update the parameter.

For x-api-key

- Sign in to your AWS account and open the AWS Management Console.
- Navigate to the API Gateway service.
- In the left-hand menu, click on "APIs".
- Click on the desired API Gateway from the list.
- In the API Gateway dashboard, click on the "API Keys" section in the left-hand menu.
- Click on the "Create API Key" button.
- In the "Name" field, enter a unique name for the API key.
- Under "API Key Source", select "Auto Generate" to let AWS automatically generate the API key value.
- Under "Usage Plans", select the appropriate usage plan for the API key.
- Click the "Save" button to create the API key.
- In the API Gateway dashboard, click on the "Resources" section in the left-hand menu.
- In the resources list, select the top-level resource (usually denoted with a forward slash "/").
- Click on the "Actions" button above the resources list and select "Deploy API".
- In the "Deployment stage" section, choose the desired stage (e.g., "prod").
- Click the "Deploy" button to redeploy the API Gateway with the new API key and configuration.