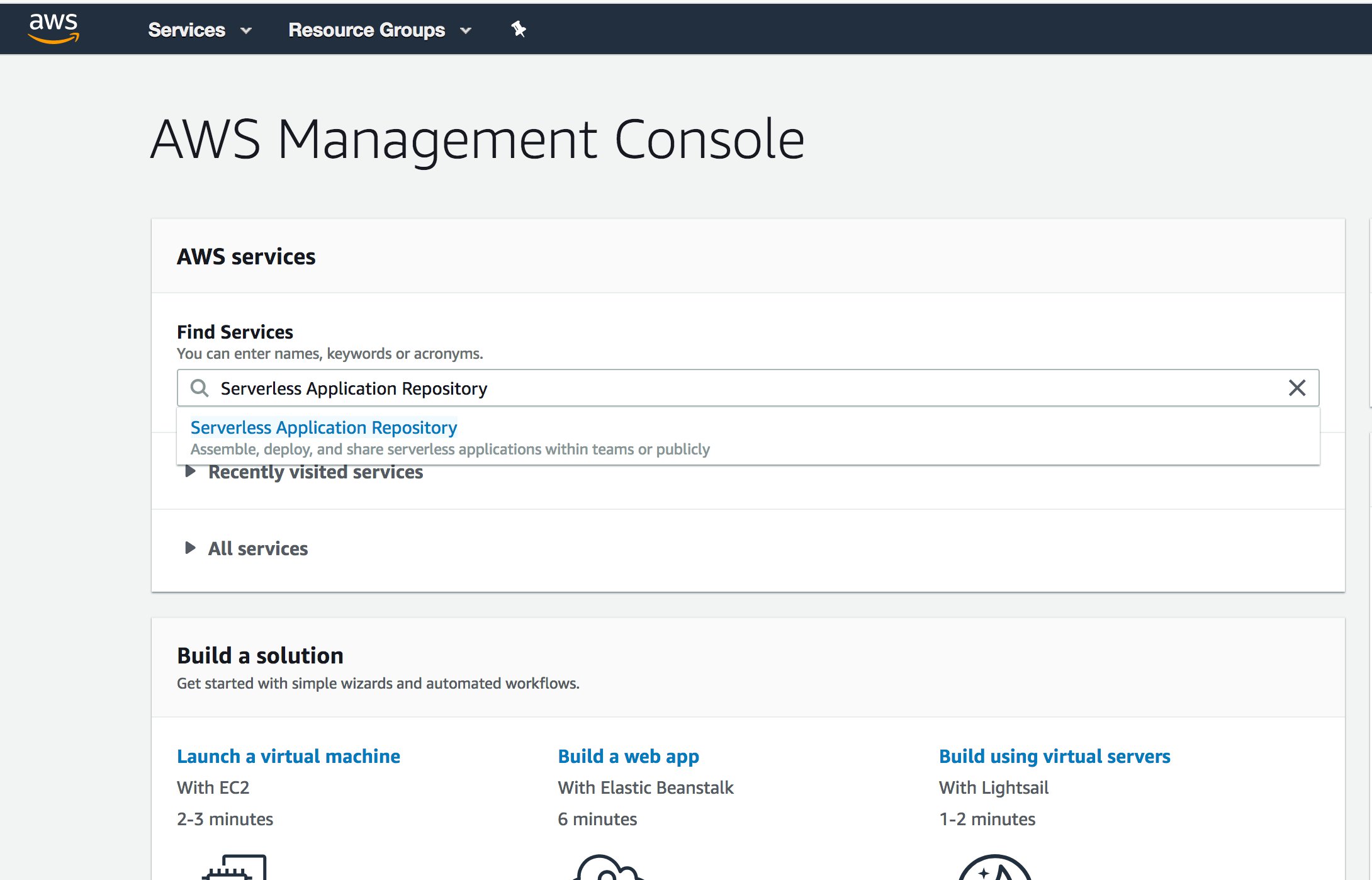
# Setting up and running Ground Truth jobs

We will be creating a Ground Truth job with a custom workflow, consisting of an HTML interface and pre-processing & post-processing Lambda functions. The HTML interface provides the human labelers with all of the instructions and tools they need to complete the labeling task. The pre-processing Lambda helps to customize input to HTML interface. The post-processing Lambda function is an accuracy improvement algorithm to tell Ground Truth how it should assess the quality of labels that humans provide.

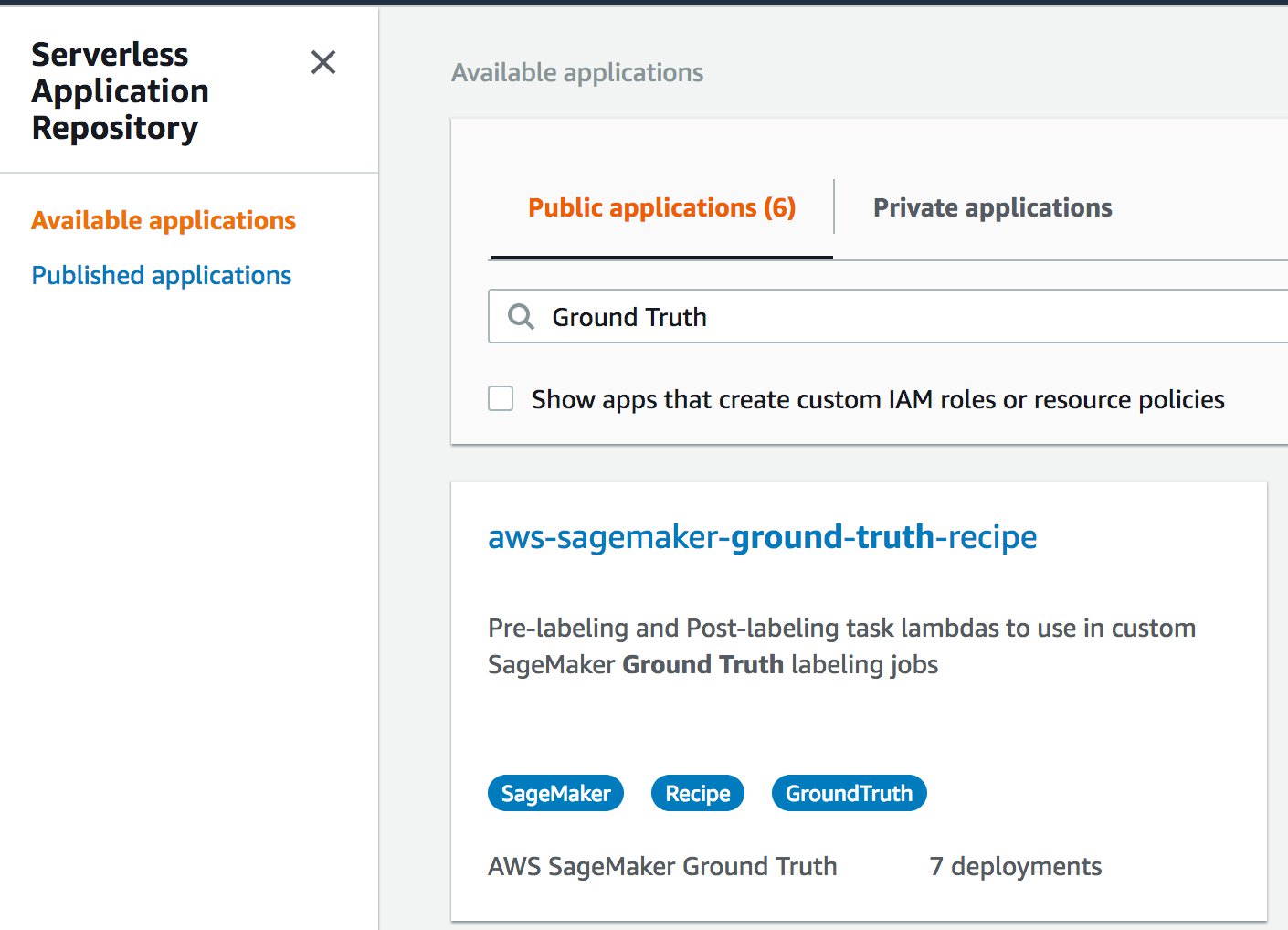
First, you need to import Lambda functions from Serverless Application Repository to your AWS account (this only has to be done once.)

## Import Lambda functions

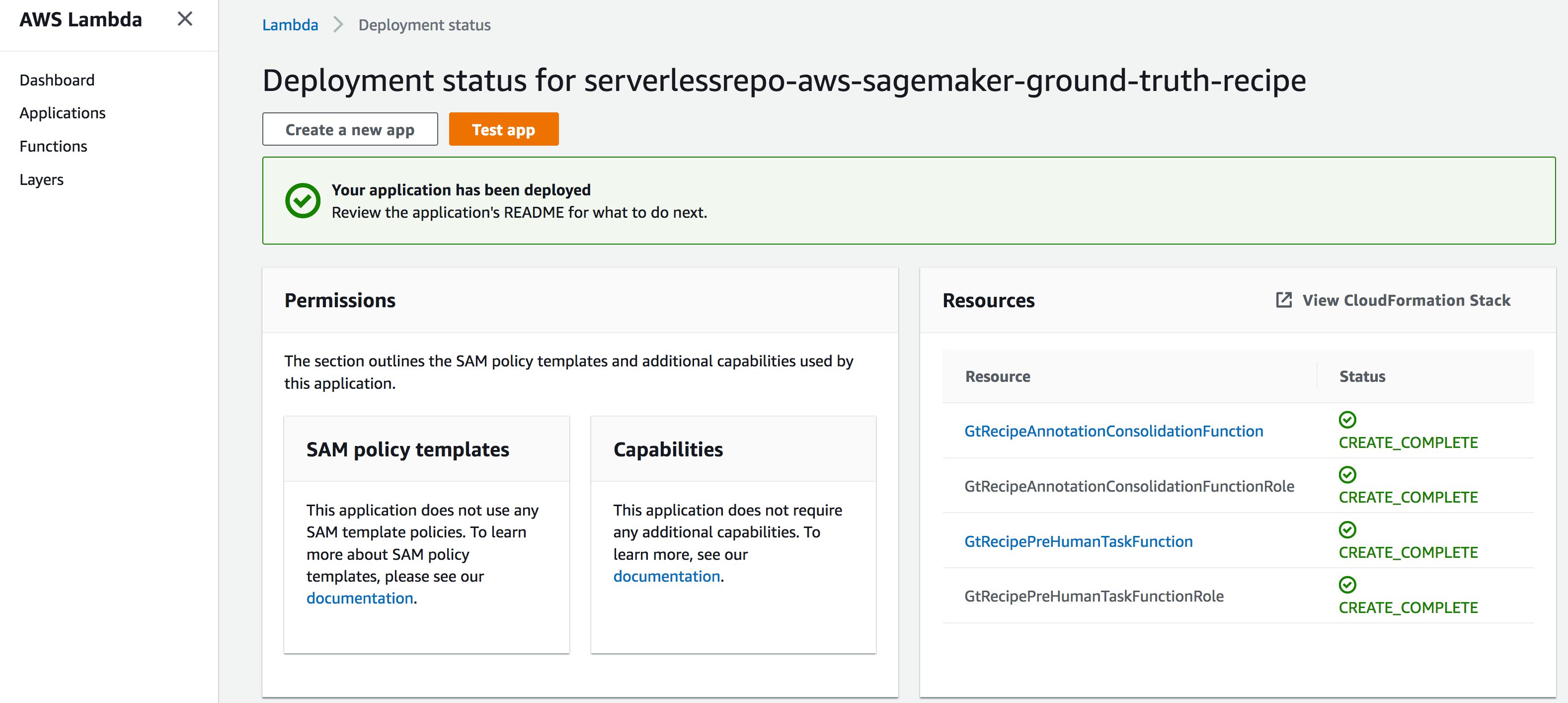
* In your AWS Management Console, search for “Serverless Application Repository”.



* On the menu on left hand side, click on “Available applications”. Search for “Ground Truth”. Click on “aws-sagemaker-ground-truth-recipe”.



* Now you will see following screen. Scroll Down to the bottom to find “Deploy” button and click on it.
* Deployment of this application may take few minutes. Please wait until you see following screen. Following status screen shows that 4 AWS resources ( 2 Lambdas and 2 Roles ) got created.



Now you have custom lambdas, to be used in labeling job, imported successfully in your account. If you would like to modify these lambdas you can do so by to clicking on the lambdas. These lambdas are written in python, so you can easily tweak them in AWS console.

## Prepare your data

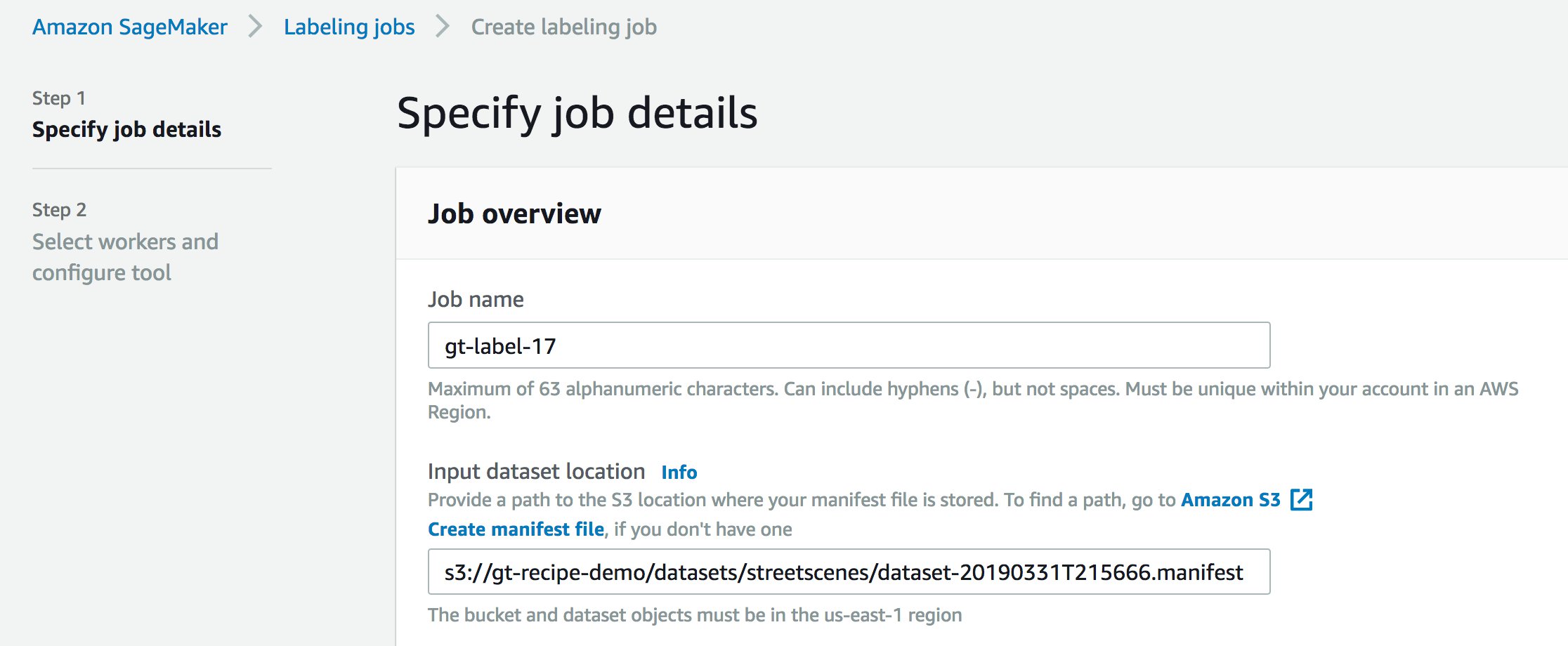
### Step 1: Upload the data to be annotated

Go to S3 in the management console, create a new bucket, and upload your data there.

### Step 2: Creating an input manifest

The input manifest is a json file that tells Ground Truth which files to load from your S3 bucket.

In the Amazon SageMaker console for Ground Truth, there is a crawling tool (see the “create manifest file” link in the input to labeling job) you can use for Ground Truth. This tool helps us create the manifest by crawling an Amazon S3 location containing raw data (image or text). For images, the crawler takes an input s3Prefix and crawls all of the image files (with extensions .jpg, .jpeg, .png) in that prefix and creates a manifest with each line as {“source-ref”:”<s3-location-of-crawled-image>”}. In the Amazon SageMaker console, start the process by creating a labeling job. First choose Labeling jobs in the left navigation pane, and then choose the Create labeling job button: Next choose Create manifest file.



This opens the create manifest file page. Enter the s3 path that you uploaded the images files to (be sure to include the trailing slash)*.* Next choose Create and then Use this manifest. (It will take a few seconds to create the manifest.)

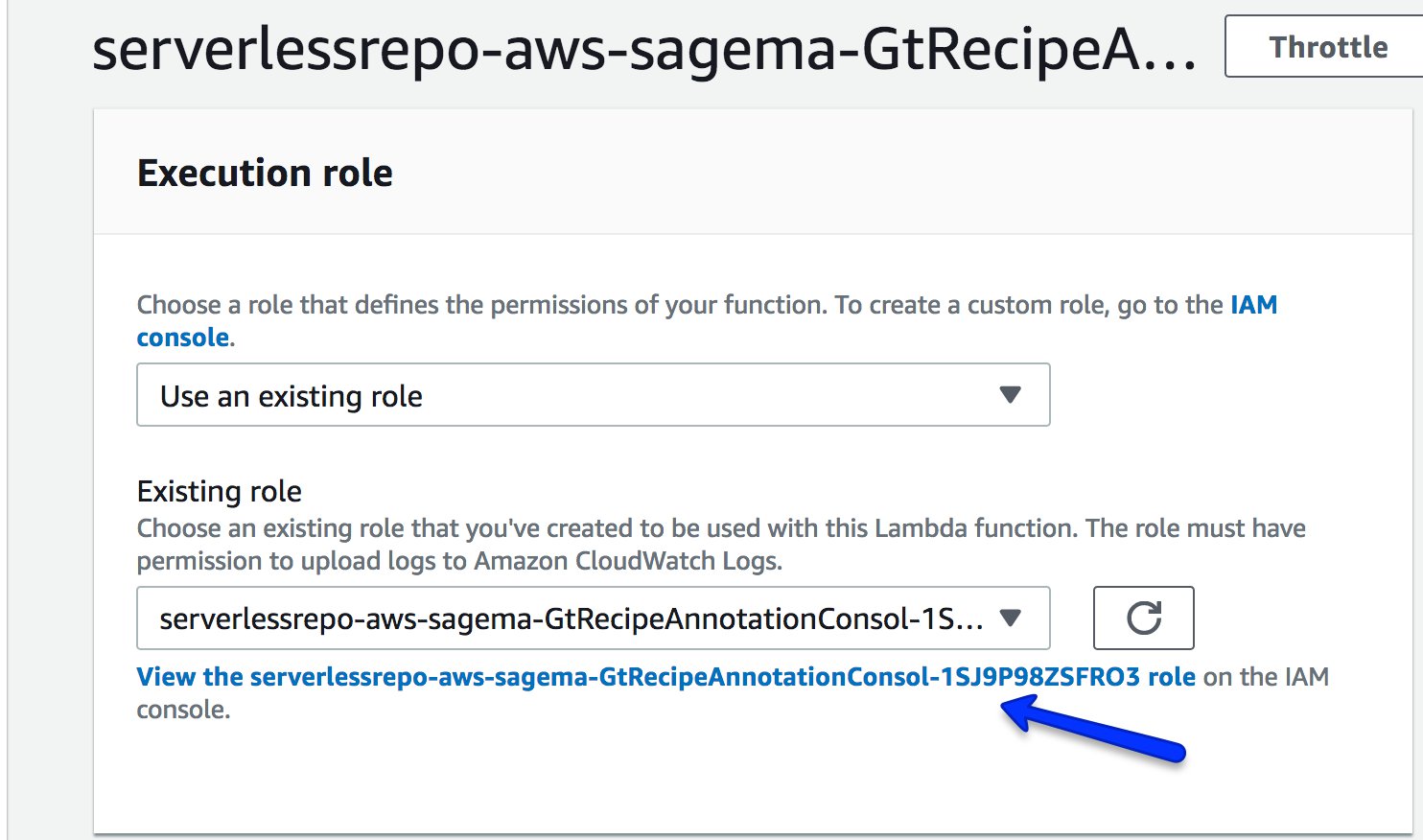
For our this example, the objects are images in Amazon S3, so we can use the crawling to create the initial manifest with each line of JSON containing a field “source-ref” pointing to the s3Uri of an image.  Content of manifest file is as follows.

{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00010.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00016.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00017.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00018.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00019.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00021.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00025.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00029.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00038.JPG"}  
{"source-ref":"s3://gt-recipe-demo/datasets/streetscenes/SSDB00039.JPG"}

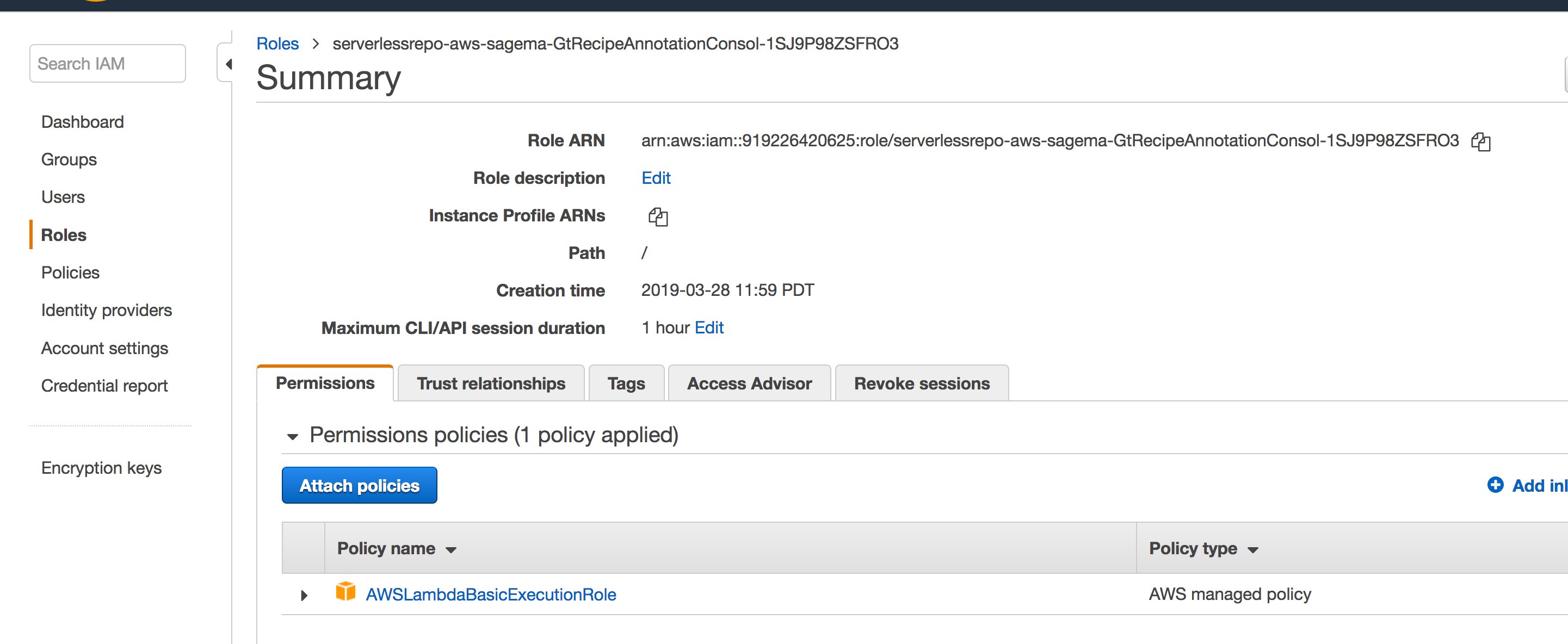
## Submitting your job

### Step 1: Give proper permission to SageMaker role to invoke Post-processing Lambda

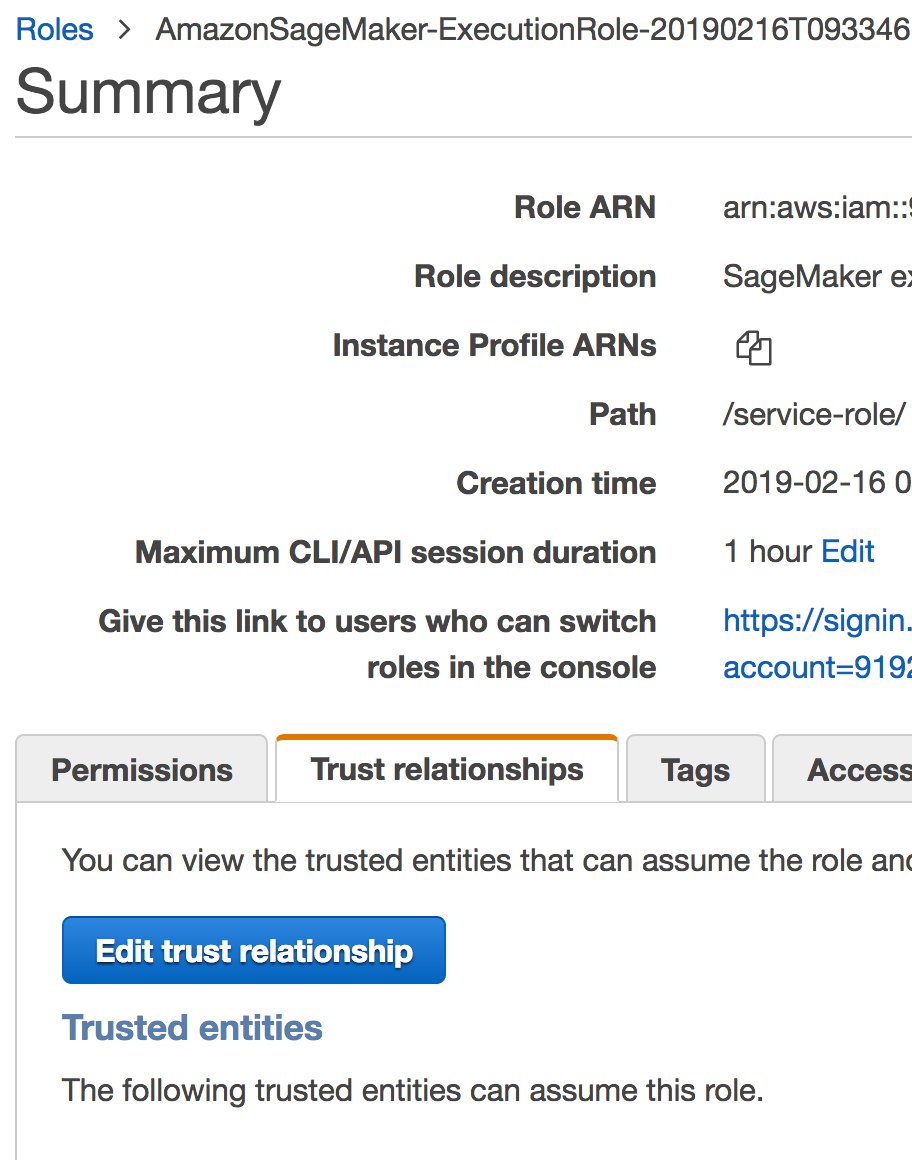
* Open a new tab on your browser. Lets call this browser tab as “role tab”
* On the newly opened tab, open AWS console. Navigate to the Lambda function that you previously created/imported. The ARN should look like this “arn:aws:lambda:us-east-1:919226420625:function:serverlessrepo-aws-sagema-GtRecipeAnnotationConsol-xxxxxxx”. Find the Execution Role, click on the role as shown below.



* Copy the Role ARN.



* Navigate to  SageMaker Execution Role,  Click on “Trust relationships” tab.



* Click on “Edit trust relationship”.  Add the copied Lambda execution role to the trust relationship. Following is the sample contents of Trust Relationship

{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": [  
          "arn:aws:iam::<your-aws-account>:role/serverlessrepo-aws-sagema-GtRecipeAnnotationConsol-xxxxx"  
        ],  
        "Service": [  
          "lambda.amazonaws.com",  
          "sagemaker.amazonaws.com"  
        ]  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}

### Step 2: Fill in job variables in the Jupyter notebook

In Amazon SageMaker, select “Notebook Instances” from the left-hand menu, and click the “Create notebook instance” button. Give it a name, keep “notebook instance type” and everything else as is.

Under “git repositories”, select “Clone a public Git repository to this notebook instance only” and clone this repo.

Finally, click “Create notebook instance”.

Back in the notebook instance menu, select your newly created instance and click “Start” (on the right). Once the instance is started, open the Jupyter notebook run\_labeling\_job.ipynb, and make the following modifications:

Specify the name of your S3 bucket containing the data to be annotated as **BUCKET**, with the manifest file inside that bucket as **MANIFEST**.

Modify the \*.template file in the annotation\_interface folder of this github repo to suite your task. Upload this file to an S3 bucket (along with any instructional images you want to appear in the instructions), and paste its path into the Jupyter notebook for the variable **UITEMPLATE**.

Change **task\_description**, **task\_keywords**, **task\_title**, and **job\_name** as desired.

Set the payment per image.

Set **prehuman\_arn** to the pre-processing ARN you imported earlier, and **acs\_arn** to the post-processing ARN. I thiiiink **workteam\_arn** can stay the way it is, but I’m not sure.

And then you can run the whole notebook! (Be sure to stop this instance later, after the job is done.)

Under Ground Truth/labeling jobs, you should see a new job that’s just been created. Once its status goes to “Complete”, the output of this job can be found in a new S3 bucket “(name of bucket with your data)-output”.