AWS Lambda

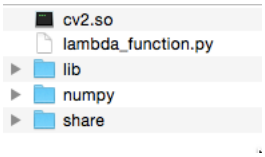
Tutorial

Face Detection

Input

Output

Sources



OpenCV statically bind library

Libraries used by numpy

numpy: Python library for library bindings

OpenCV configuration files

Face Detection

```
import ...

print('Loading function')

dstBucket = 'ece1779winter2017demo'
dstPrefix = 'tmp/'
outputDomain = 's3.amazonaws.com/ece1779winter2017demo'

cascPath = 'share/OpenCV/haarcascades/haarcascade_frontalface_alt.xml'

s3 = boto3.resource('s3')
faceCascade = cv2.CascadeClassifier(cascPath)

def lambda_handler(event, context):
    print('Received event: ' + json.dumps(event, indent=2))
    imageUrl = event['imageUrl']
    imageFile = urllib2.urlopen(imageUrl)
    imageBytes = numpy.asarray(bytearray(imageFile.read()), dtype=numpy.uint8)
    image = cv2.imdecode(imageBytes, cv2.CV_LOAD_IMAGE_UNCHANGED)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags = cv2.cv.CV_HAAR_SCALE_IMAGE
    )

    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h), (255, 255, 255), 2)

    r, outputImage = cv2.imencode('.jpg', image)
    if False==r:
        raise Exception('Error encoding image')

    dstKey = dstPrefix + datetime.datetime.now().strftime('%Y%m%d%H%M%S') + '-' + str(uuid.uuid4()) + '.jpg'

    s3.Bucket(dstBucket).put_object(Key=dstKey,
        Body=outputImage.tostring(),
        ContentType='image/jpeg',
        ACL='public-read'
    )

    outputUrl = 'https://' + outputDomain + '/' + dstKey
    return "<html><body><img src='{}' /></body></html>".format(outputUrl)
```

Replace with YOUR S3 bucket name

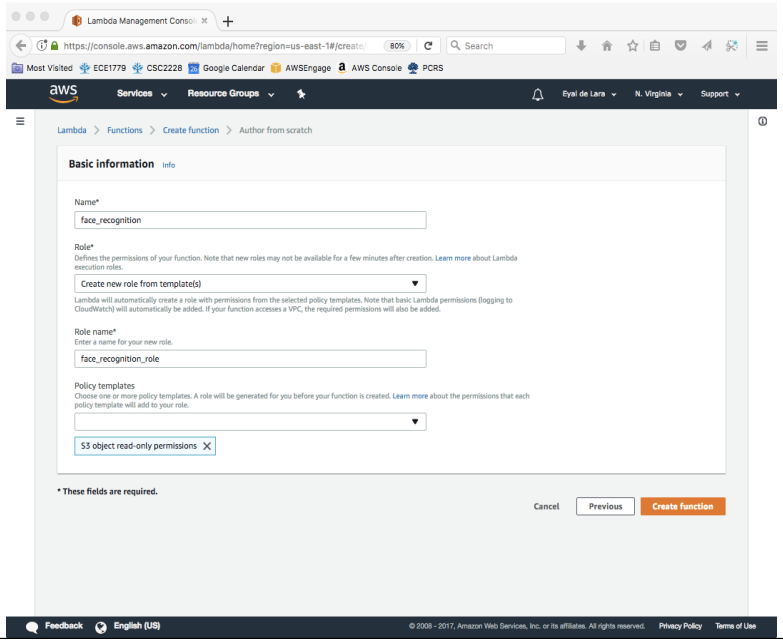
Replace with URL to your S3 bucket

Create Deployment Package

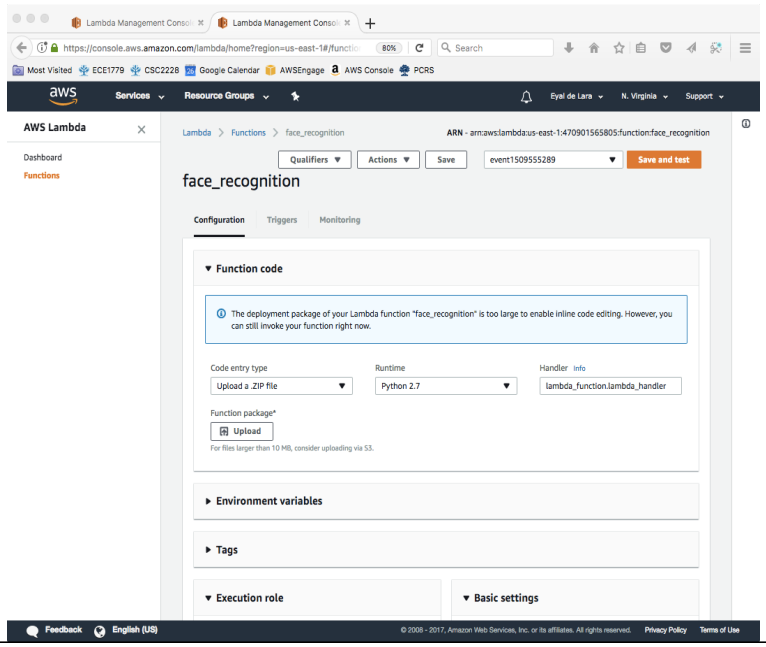
- **Inside source directory type:**

  > zip –r face_detection_code.zip *

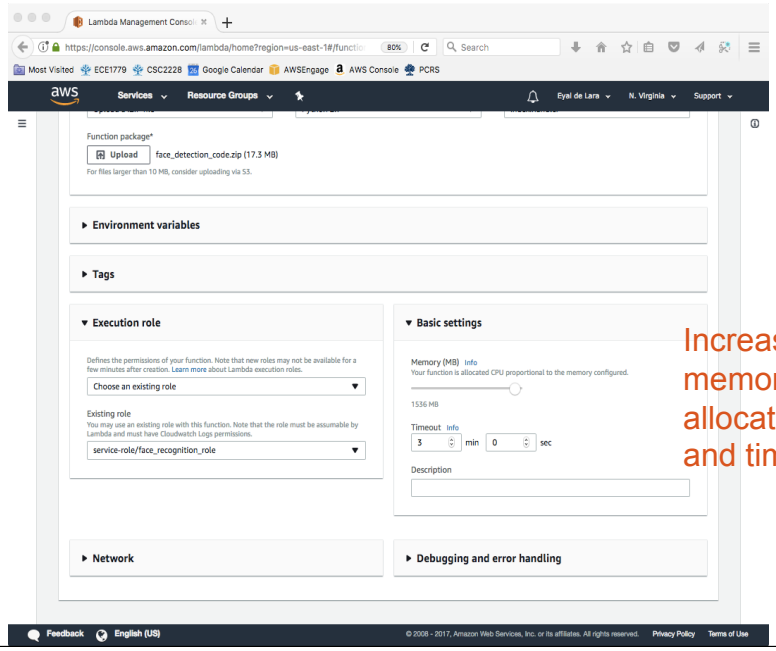
Create Lambda Function



Create Lambda Function

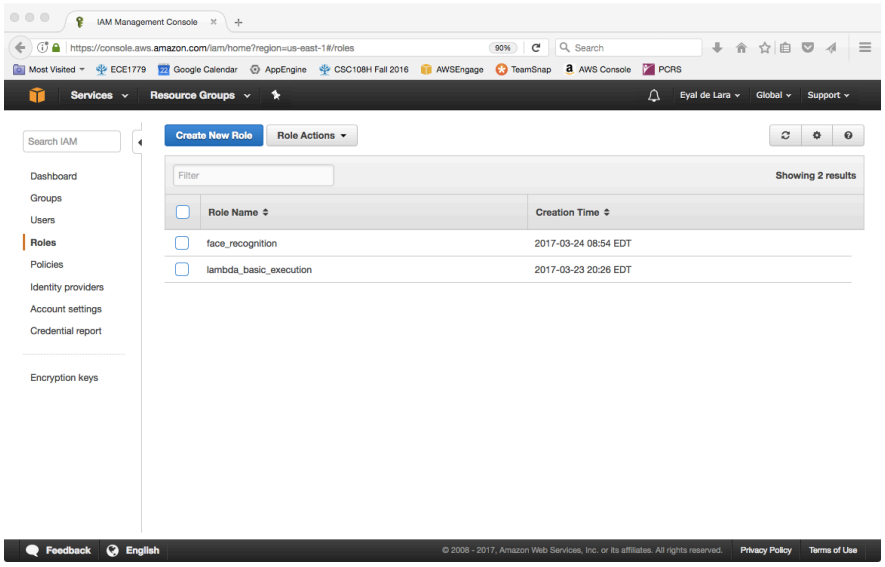


Create Lambda Function

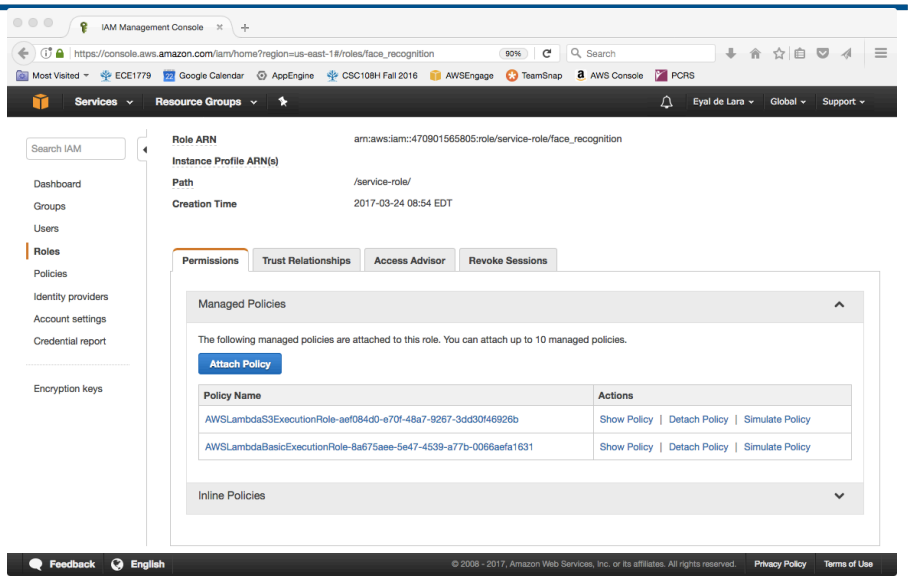


Increase memory allocation and timeout

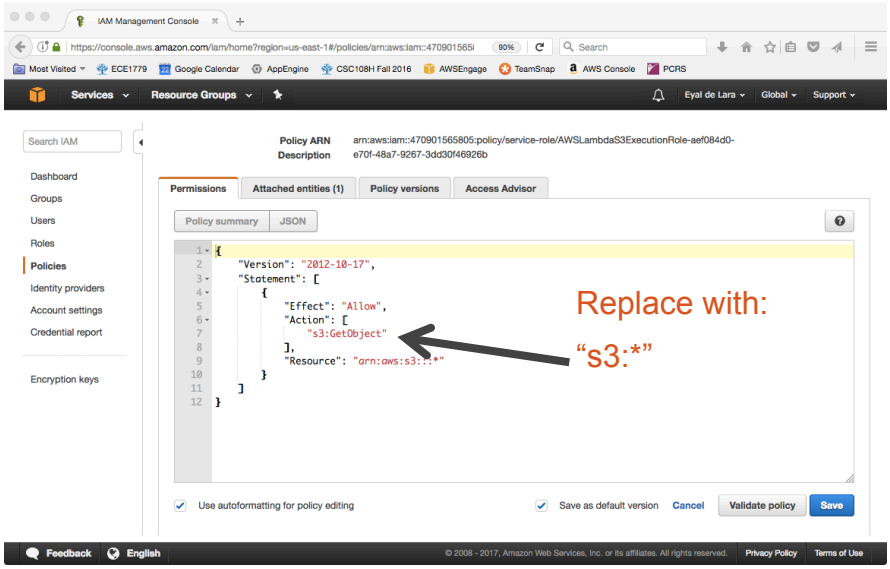
Configure Security Role



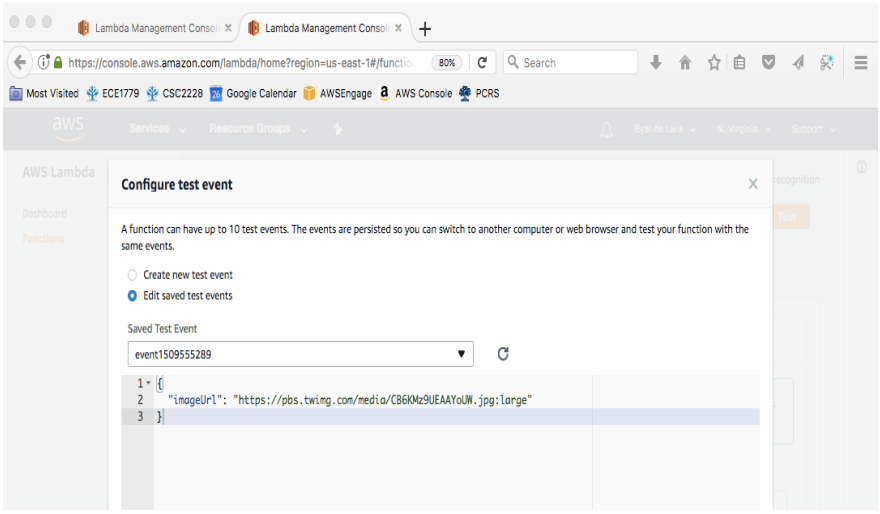
Configure Security Role



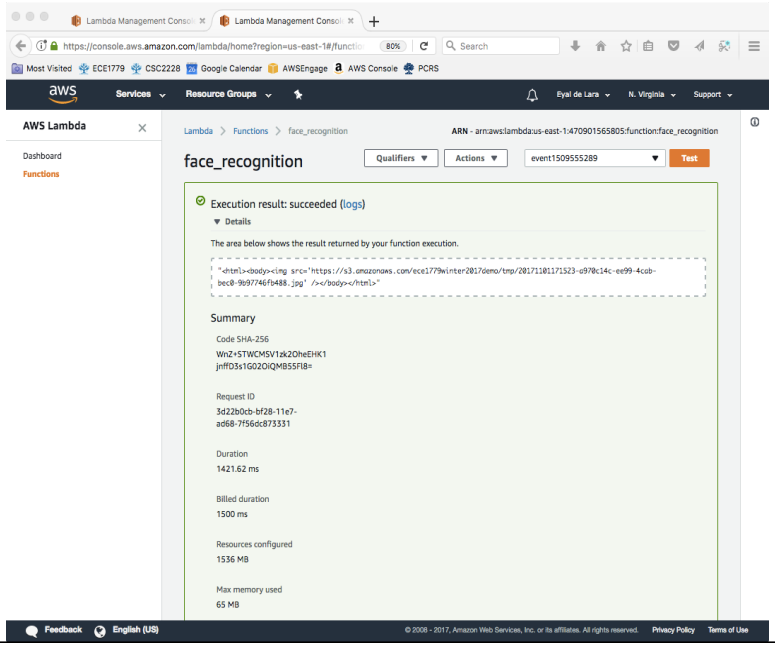
Configure Security Role



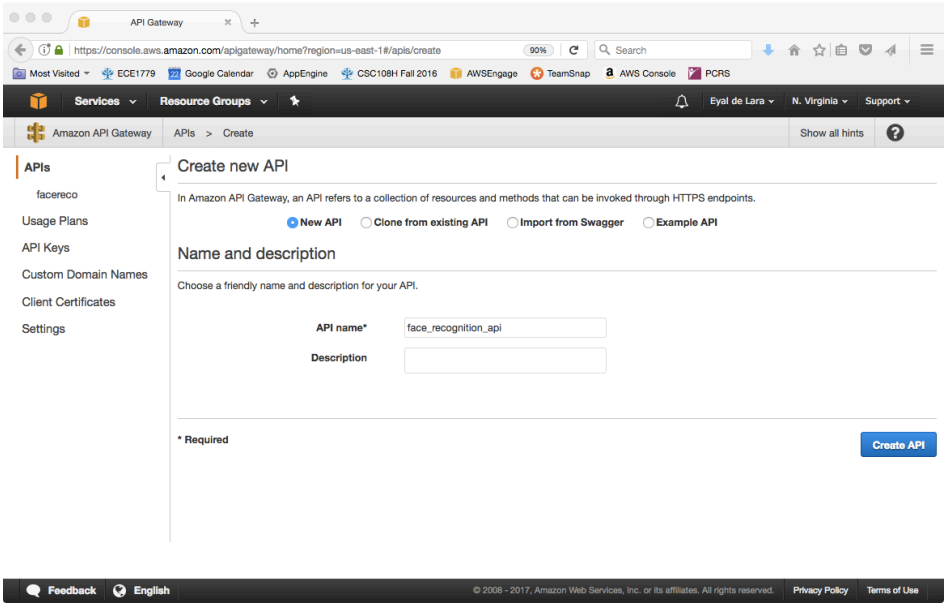
Configure Test Values



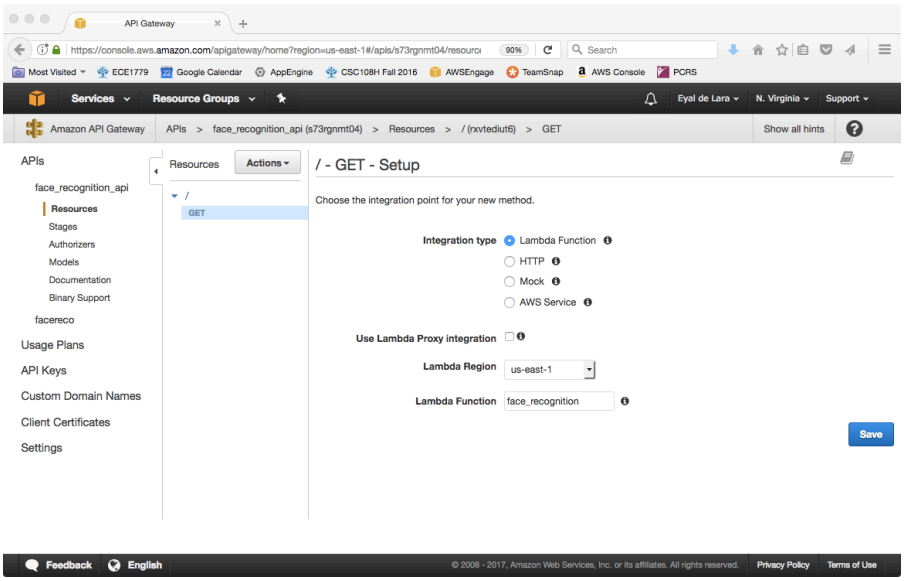
Test Function



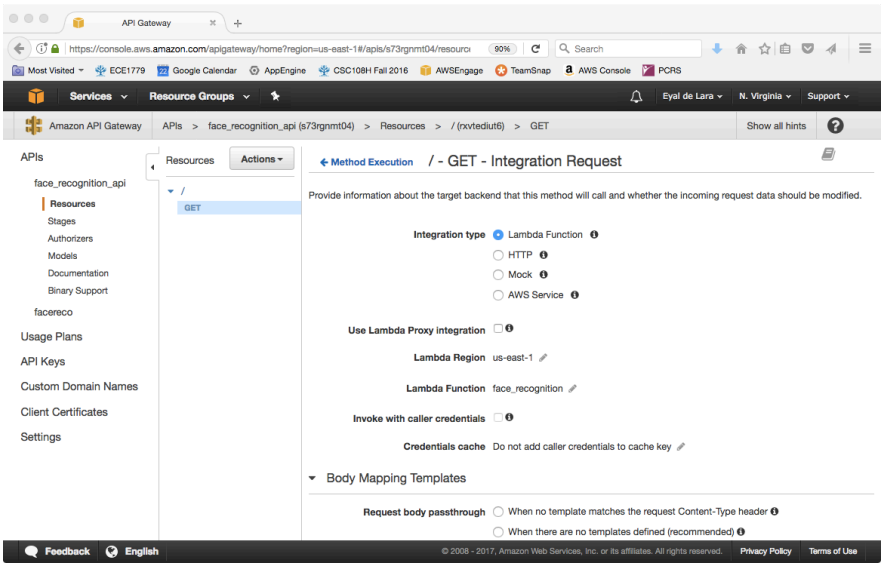
Create Web API



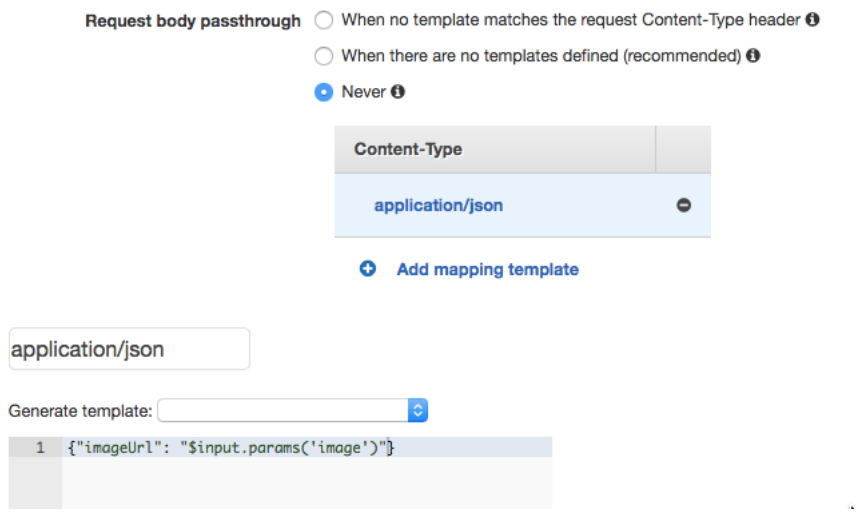
Create Method



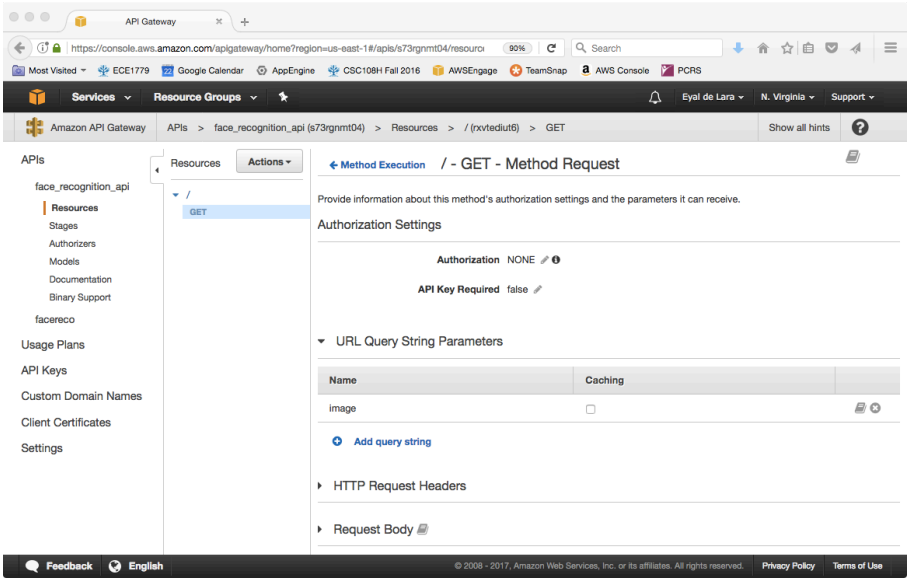
Map HTTP Parameter for Function Argument



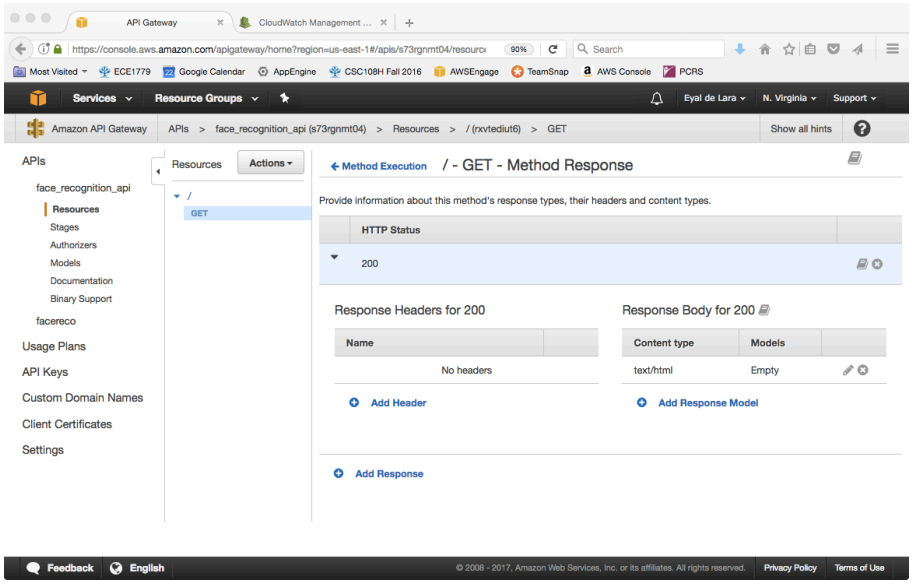
Map HTTP Parameter for Function Argument



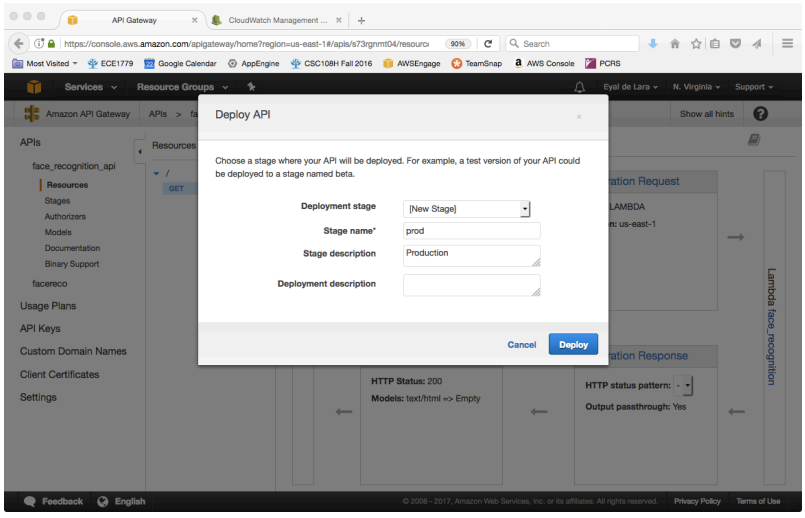
Add Parameter to Method Request (for testing purposes)



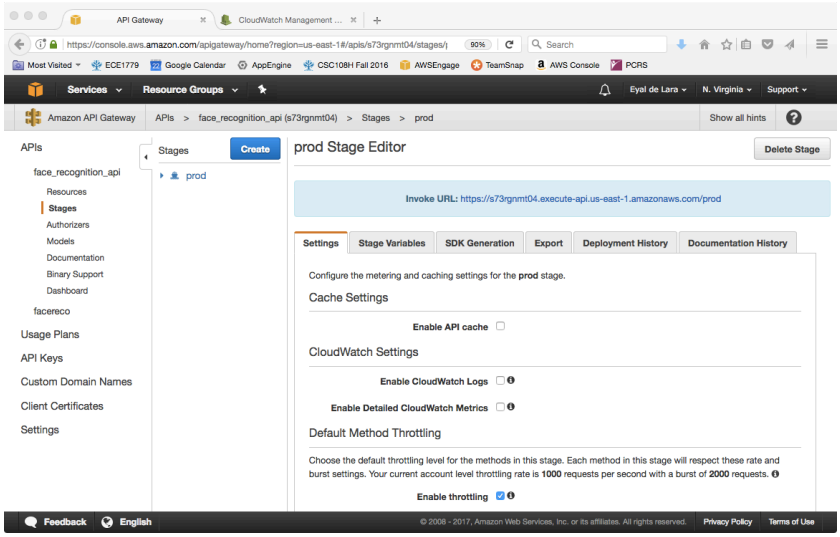
Change Method Response Type to HTML



Deploy API



Deploy API



Test with Browser