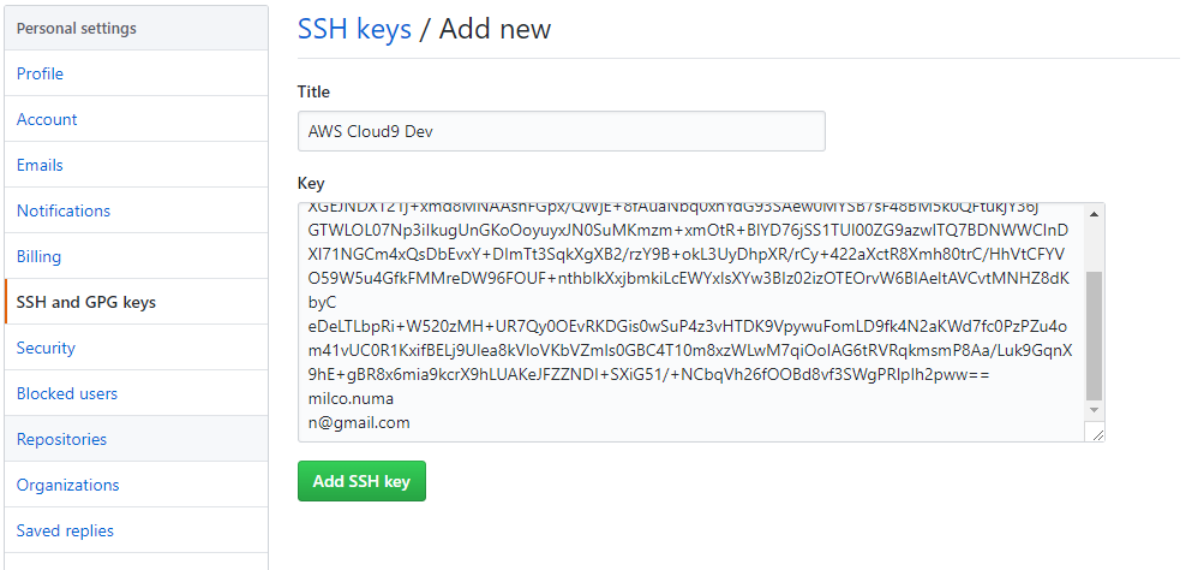
Guys don’t want to commit!

1) Use your current account at twitter or register a new account
2) Register a new app on this Twitter account, you’ll need:
    a. Consumer key (API key)
    b. Consumer secret (API secret)
    c. Access token (generate)
    d. Access token secret (generated with the access token)
3) Create a new empty repository in Github (note the URL)
4) Create a new empty repository on your Cloud9 instance (from a terminal window):

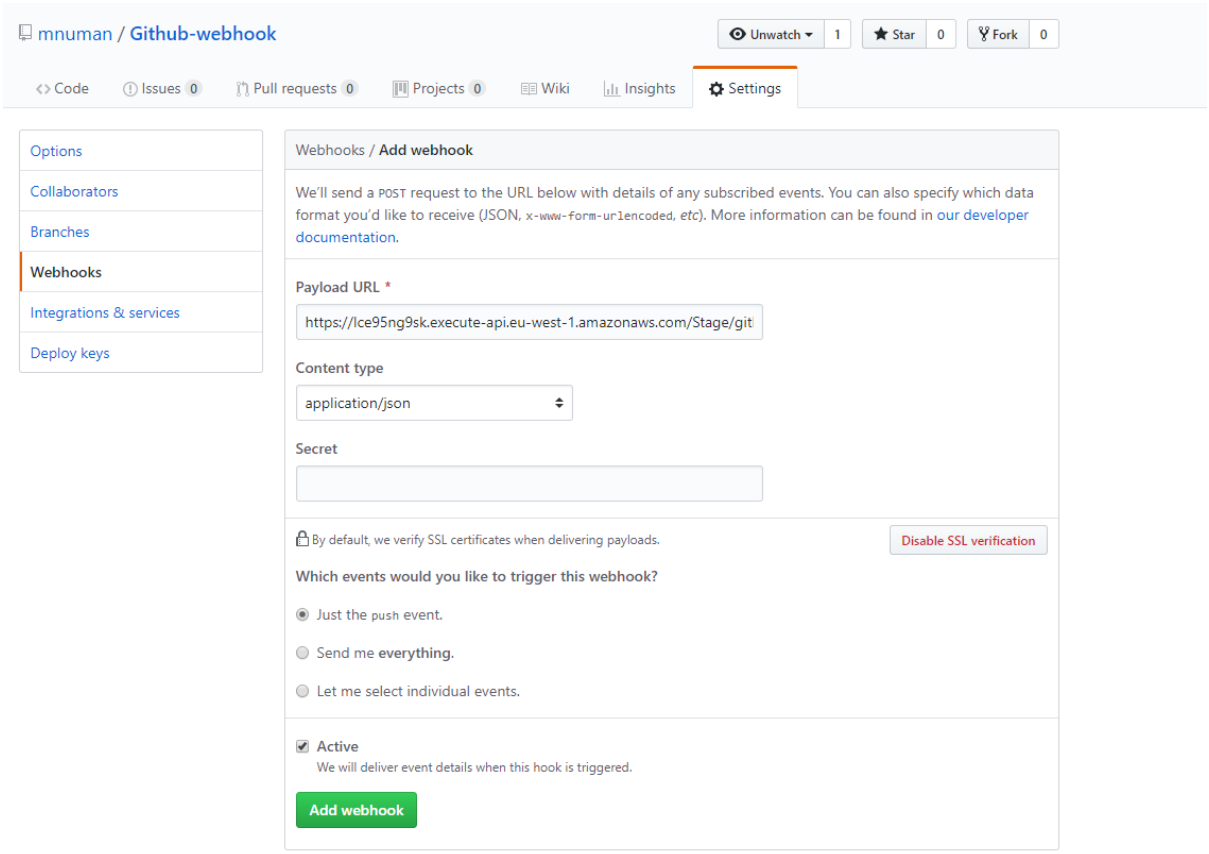
```
git init .
git remote add origin git@github.com:<your-github-username>/<Your-Github-Reponame>
```

5) Generate a new keypair from your Cloud9 terminal window:
   ssh-keygen -t rsa -b 4096 -C “**your-email-for-github**”
6) Configure your git client by setting your identity for git (Cloud9, terminal window):
   git config --global user.name “User name to show in github”
   git config – global user.email “your-github-email-here”
7) Now ad d the public key (~/.ssh/id_rsa.pub) you generated in step 6 to your Github account:



8) Create a new serverless application + function in Cloud9, type empty python 3.6, triggered by API Gateway on resource path /github, no security. Defaults apply.
9) Deploy the API to AWS to get an URL for the API from API gateway

10) Navigate to Github, add a new webhook to the repository you created for this purpose:



11) Test from Github that the invoke works
12) Now commit your code from Cloud9's terminal window and push this to Github. This should trigger the actual lambda function.
13) Verify in CloudWatch that your lambda function has been triggered and inspect the payload.
14) Change into your Application's directory using the Cloud9 terminal window and install the Twitter package here locally:
    `pip install tweepy -t .`
15) Generate the requirements file and prevent the dependencies from being checked in by editing the .gitignore file:

```
milco:~/environment/GithubWebhook (master) $ pip freeze > requirements.txt
milco:~/environment/GithubWebhook (master) $ more .gitignore
.application.json
certifi/
chardet/
examples/
idna/
oauthlib/
requests/
requests_oauthlib/
tweepy/
urllib3/
milco:~/environment/GithubWebhook (master) $
```

16) Add the secrets/keys as configuration variables to your configuration template (the yaml template):

template.yaml

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: 'AWS::Serverless-2016-10-31'
Description: An AWS Serverless Specification template describing your function.
Resources:
  GithubWebhook:
    Type: 'AWS::Serverless::Function'
    Properties:
      Handler: GithubWebhook/lambda_function.lambda_handler
      Runtime: python3.6
      Description: 'Tweet on commit'
      MemorySize: 128
      Timeout: 15
      Environment:
        Variables:
          CONSUMER_KEY: 5QaNYVeUSx6L4EtSqe50mR8SQ
          CONSUMER_SECRET: P1io6NFS4Ql93pRJOmzXZeH16CAKUZ9MSiuKEYVDWFeIBR4gvC
          ACCESS_TOKEN: 966365883114409984-DPLfbbqr5biFLcXNqZjXIgbZsJZ37rq
          ACCESS_TOKEN_SECRET: t7gEDs5ecCevXUaeLGcLZ7CeZ2cFGtDoxp34svpaEQ9yb
      Events:
        LambdaMicroservice:
          Type: Api
          Properties:
            Path: /github
            Method: ANY
  GithubWebhookPermission:
    Type: 'AWS::Lambda::Permission'
    Properties:
      Action: 'lambda:InvokeFunction'
      FunctionName:
        'Fn::GetAtt':
          - GithubWebhook
          - Arn
      Principal: apigateway.amazonaws.com
      SourceArn:
        'Fn::Sub': 'arn:aws:execute-api:${AWS::Region}:${AWS::AccountId}:*/*/*/*'

```

17) Finish the implementation code in Python:

lambda_function.py

```python
import json
import os
import tweepy

# importing the environment variables - is this safe? We'll see ...
consumer_key = os.environ['CONSUMER_KEY']
consumer_secret = os.environ['CONSUMER_SECRET']
access_token = os.environ['ACCESS_TOKEN']
access_token_secret = os.environ['ACCESS_TOKEN_SECRET']

# setup authentication with twitter
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

def lambda_handler(event, context):
    body = json.loads(event['body'])
    mytweet = f"Hi, {body['head_commit']['committer']['name']} just commited with message {body['head_commit']['message']}!"
    status = api.update_status(status=mytweet)

    response_body =  { "message": "Your tweet has been posted!"}

    return {
      "isBase64Encoded": "false",
      "statusCode": 200,
      "headers": {},
      "body": json.dumps(response_body)
    }
```

bash - "ip-172-31-44

```
milco:~/environment (master) $ clear
milco:~/environment (master) $ git add .
milco:~/environment (master) $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   GithubWebhook/GithubWebhook/lambda_function.py

milco:~/environment (master) $ git commit -m "Take author/message from head_commit instead of commits array"
[master 9b5697c] Take author/message from head_commit instead of commits array
 1 file changed, 1 insertion(+), 1 deletion(-)
milco:~/environment (master) $ date ; git push
Sun Feb 25 14:16:58 UTC 2018
Counting objects: 5, done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 520 bytes | 520.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:mnuman/Github-webhook.git
   e877214..9b5697c  master -> master
milco:~/environment (master) $
```

18) Now deploy the API again, since the code has changed since last push.
19) Finish up by committing your changed code to Github
20) Now you should see a message on twitter: