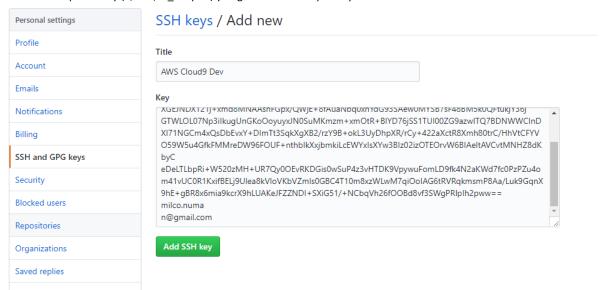
## Guys don't want to commit!

- 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):

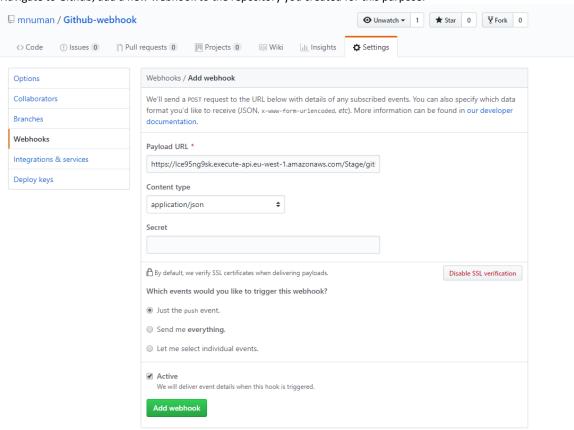
```
git init .
git remote add origin git@github.com:<your-github-username>/<Your-Github-
Reponame>
```

- 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 x +

AMSTemplateFormatVersion: '2010-09-09'
Transform: 'AMS::Serverless-2016-10-31'
Description: An AMS Serverless Specification template describing your function.
Resources:
GithubMebhook:
Type: 'AMS::Serverless::Function'
Properties:
Handler: GithubMebhook/lambda_function.lambda_handler
Runtime: python3.6
Description: 'Tweet on commit'
HemorySize: 128
Timeout: 15
Timeout: 15
CONSUMER EK: SQMYVEUSX6L4EtSqe50mR8SQ
CONSUMER EK: SQMYVEUSX6L4EtSqe50mR8SQ
CONSUMER EK: PidoGMF54Q103pR3OmzXZeH16CAKUZ9MSiuKEYVDMFeIBR4gvC
ACCESS_TOKEN: 966365883114409984-DPLfbbqr5biFLcXNqzjXtgbzs1237rq
ACCESS_TOKEN: SECRET: t7gEDS5ecCevXUJaeLGcLZ7CeZ2cFGtDoxp34svpaEQ9yb
Events:
LambdaMicroservice:
Type: Api
Properties:
Path: /github
Mcthod: AMY
GithubMebhookPermission:
Type: 'AMS::Lambda::ProvkeFunction'
FunctionName:
Fin::GetAtt':
- GithubMebhook
- Arn

Fin::GetAtt':
- GithubMebhook
- Arn

Principal: apigateway.amazonaws.com
SourceArn:
'Fn::Sub': 'arn:aws:execute-api:${AMS::Region}:${AMS::AccountId}:*/*/**
```

17) Finish the implementation code in Python:

- 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:



Hi, Milco Numan just commited with message Take author/message from head\_commit instead of commits array!

