

## 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 add the public key (`~/.ssh/id_rsa.pub`) you generated in step 6 to your Github account:

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

Saved replies

### SSH keys / Add new

Title

AWS Cloud9 Dev

Key

XGEJNDX1Z1J+Xmd8MINAAsnFgpx/QVWJE+8TAuaInbquxnYdG93SAEWUMY5B/5f48Bm5KUQFtUKJY3bJ  
GTWLOLO7Np3ilkugUnGKoOoyuyxjN0SuMKmzm+xmOtr+BIYD76jSS1TUI00ZG9azwITQ7BDNWWCInD  
XI71NGCm4xQsDbEvxY+DImTt3SqkXgXB2/rzY9B+okL3UyDhpXR/rCy+422aXctR8Xmh80trC/HhVtCFYV  
O59W5u4GfkFMMreDW96FOUF+nthblKxjbmkiLcEWYxIsXYw3Blz02izOTEOrvW6BIAeltAVCvtMNHZ8dK  
byC  
eDeLTlpRi+W520zMH+UR7Qy0OEVRKDGis0wSuP4z3vHTDK9VpywuFomLD9fk4N2aKWd7fc0PzPZu4o  
m41vUC0R1KxifBELj9Ulea8kVloVKbVZmls0GBc4T10m8xzWLwM7qiOolAG6tRVrqkmsmP8Aa/Luk9GqnX  
9hE+gBR8x6mia9kcrX9hLUAKeJFZZNDI+SXIG51/+NCbqVh26fOOBd8vf3SWgPRlplh2pww==  
milco.numa  
n@gmail.com

Add SSH key

- 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:

The screenshot shows the GitHub 'Add webhook' page for the repository 'mnuman / Github-webhook'. The left sidebar contains navigation links: Options, Collaborators, Branches, Webhooks (highlighted), Integrations & services, and Deploy keys. The main content area is titled 'Webhooks / Add webhook' and includes the following fields and options:

- Payload URL \***: `https://lce95ng9sk.execute-api.eu-west-1.amazonaws.com/Stage/git`
- Content type**: `application/json`
- Secret**: (Empty text input field)
- SSL verification**: A checkbox labeled 'By default, we verify SSL certificates when delivering payloads.' with a 'Disable SSL verification' button.
- Which events would you like to trigger this webhook?**: Three radio button options: 'Just the push event.' (selected), 'Send me everything.', and 'Let me select individual events.'
- Active**: A checked checkbox with the text 'We will deliver event details when this hook is triggered.'
- Add webhook**: A green button at the bottom.

- 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
1  AWSTemplateFormatVersion: '2010-09-09'
2  Transform: 'AWS::Serverless-2016-10-31'
3  Description: An AWS Serverless Specification template describing your function.
4  Resources:
5    GithubWebhook:
6      Type: 'AWS::Serverless::Function'
7      Properties:
8        Handler: GithubWebhook/lambda_function.lambda_handler
9        Runtime: python3.6
10       Description: 'Tweet on commit'
11       MemorySize: 128
12       Timeout: 15
13       Environment:
14         Variables:
15           CONSUMER_KEY: 5QaNYVeUSx6L4Et5qe50mR8SQ
16           CONSUMER_SECRET: P1io6NFS4Q193pRJ0mzXZeH16CAKUZ9MSiuKEYVDWFeIBR4gvC
17           ACCESS_TOKEN: 966365883114409984-DPLfbbqr5biFLcXNqZjXIgbZsJZ37rq
18           ACCESS_TOKEN_SECRET: t7gEDs5ecCevXUaeLGcLZ7CeZ2cFGtDoxp34svpaEQ9yb
19       Events:
20         LambdaMicroservice:
21           Type: Api
22           Properties:
23             Path: /github
24             Method: ANY
25       GithubWebhookPermission:
26         Type: 'AWS::Lambda::Permission'
27         Properties:
28           Action: 'lambda:InvokeFunction'
29           FunctionName:
30             'Fn::GetAtt':
31               - GithubWebhook
32               - Arn
33           Principal: apigateway.amazonaws.com
34           SourceArn:
35             'Fn::Sub': 'arn:aws:execute-api:${AWS::Region}:${AWS::AccountId}:*/*/*/'
36
```

17) Finish the implementation code in Python:

```
lambda_function.py x +
1 import json
2 import os
3 import tweepy
4
5 # importing the environment variables - is this safe? We'll see ...
6 consumer_key = os.environ['CONSUMER_KEY']
7 consumer_secret = os.environ['CONSUMER_SECRET']
8 access_token = os.environ['ACCESS_TOKEN']
9 access_token_secret = os.environ['ACCESS_TOKEN_SECRET']
10
11 # setup authentication with twitter
12 auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
13 auth.set_access_token(access_token, access_token_secret)
14
15 api = tweepy.API(auth)
16
17 def lambda_handler(event, context):
18     body = json.loads(event['body'])
19     mytweet = f"Hi, {body['head_commit']['committer']['name']} just committed with message {body['head_commit']['message']}!"
20     status = api.update_status(status=mytweet)
21
22     response_body = { "message": "Your tweet has been posted!" }
23
24     return {
25         "isBase64Encoded": "false",
26         "statusCode": 200,
27         "headers": {},
28         "body": json.dumps(response_body)
29     }
```

```
bash - "ip-172-31-44" x +
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:



**Milco Tweetbot**

@TweetbotMilco



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

🌐 Vertalen uit het Engels

06:17 - 25 feb. 2018

