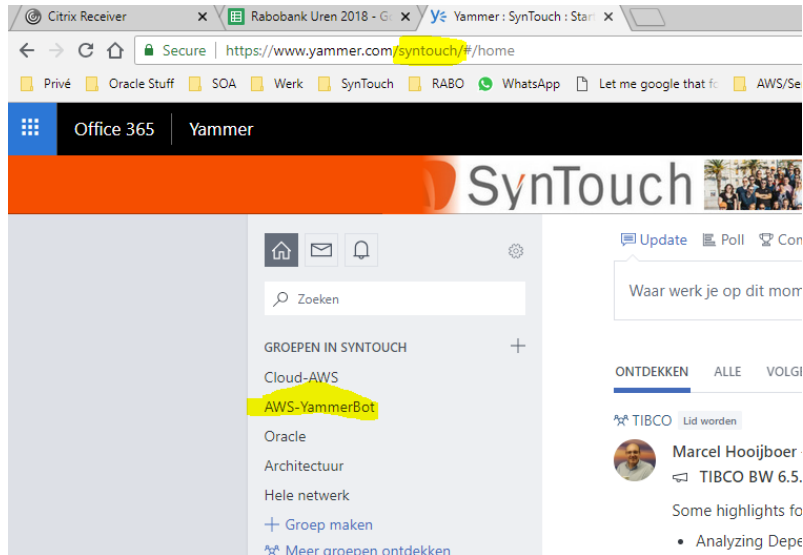


Guys don't want to commit!

- 1) Use your current Yammer-account to log into our **SynTouch** Yammer-network; if you have multiple networks on your Yammer account, please make sure to switch to the SynTouch-network:



- 2) Open a new tab on your browser to https://www.yammer.com/client_applications
- 3) Use the button to register a new application, prefix this with your name:

Nieuwe app registreren

Alle velden zijn verplicht.

Toepassingsnaam	MilcosYammerBot
Organisatie	SynTouch
E-mail voor ondersteuning	milco.numan@syntouch.nl
Website	http://www.syntouch.nl
Omleidings-URI	http://127.0.0.1

☒ Door dit selectievakje in te schakelen, geef je aan dat je de Yammer API-servicevoorwaarden hebt gelezen en hiermee akkoord gaat.

- 4) On the confirmation page, use the link to generate a developer token and capture the value:

Mijn apps

MilcoNumanYammerBot

MilcoNumanYammerBot

Sleutels en tokens

Client-id qUp1ax2s1B3Kav1L7TdW

Clientgeheim tgErULBj8FA3XIZsqeWfrnmNxSkU177TolJPMw

Verwachte omleiding http://127.0.0.1

Een developer token genereren voor deze toepassing

Nieuwe app registreren

Hulp krijgen

- API-documentatie
- Problemen met de site
- Partner met Yammer

Mijn apps

MilcoNumanYammerBot

MilcoNumanYammerBot

Sleutels en tokens

Client-id qUp1ax2s1B3Kav1L7TdW

Clientgeheim tgErULBj8FA3XIZsqeWfrnmNxSkU177TolJPMw

Verwachte omleiding http://127.0.0.1

Hier is uw persoonlijke token voor het testen. Kopieer het naar een veilige plaats, want het verdwijnt wanneer u de pagina verlaat:

10891656 42ngSZc7n40...

Nieuwe app registreren

Hulp krijgen

- API-documentatie
- Problemen met de site
- Partner met Yammer

- 5) Open a new tab in your browser and lookup the groups in your network on URL <https://www.yammer.com/api/v1/groups.json>. Find the AWS-YammerBot group and note its id
- 6) Create a new empty repository in Github (note its SSH address), use the Python gitignore option:

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

mnuman / serverless-yammer

Great repository names are short and memorable. Need inspiration? How about reimagined-potato.

Description (optional)

Repository for my serverless yammer github hook

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

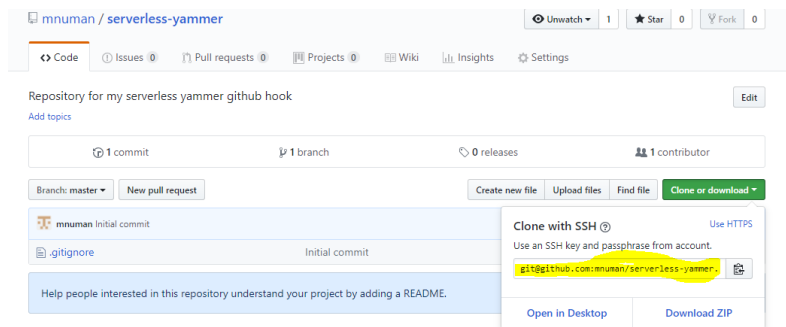
☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

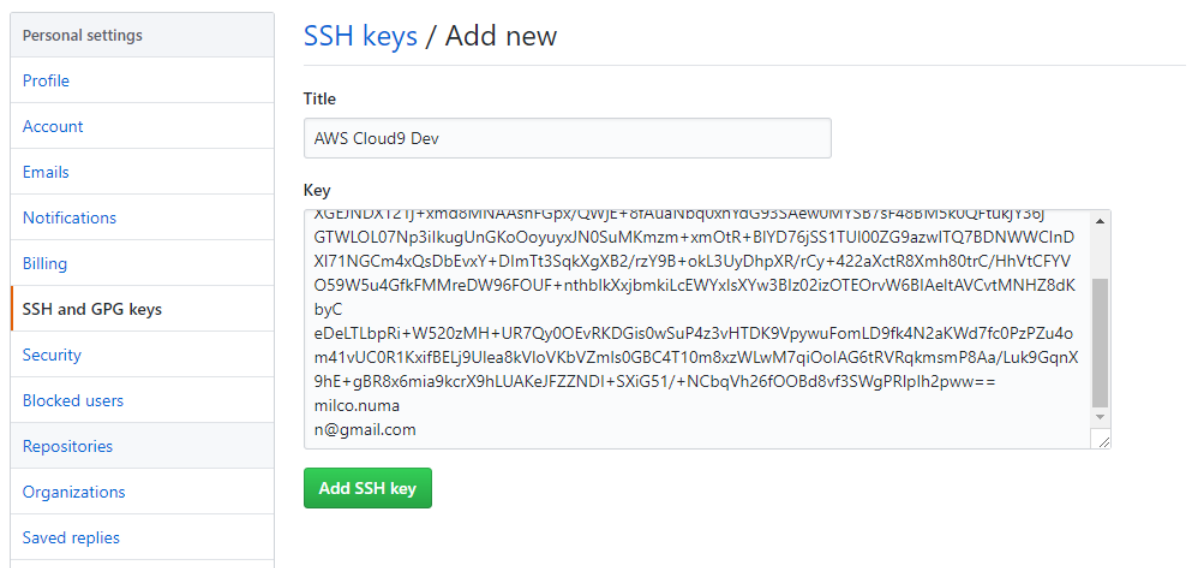
Add .gitignore: Python

Add a license: None

Create repository



- 7) Generate a new keypair from your Cloud9 terminal window:
ssh-keygen -t rsa -b 4096 -C "**your-email-for-github**"
- 8) 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"
- 9) Now add the public key (~/.ssh/id_rsa.pub) you generated in step 6 to your Github account:



- 10) In Cloud9's terminal window, move to your home directory and initialize a new git repository there:
git init .
- 11) Connect the new local repository to your GitHub repository (text in italics should be changed into YOUR repo address!):
git remote add origin git@github.com:mnuman/serverless-yammer.git
- 12) Synchronize by pulling in all remote changes and merging them into the master branch:
git pull origin master

- 13) 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:

Create serverless application ✕

Review your serverless application details below. You can go back to make changes for each section. When you are ready, click Finish to complete the setup process.

After AWS Cloud9 creates your new application, it will be automatically deployed using AWS CloudFormation.

Triggers

api-gateway

Resource Path /github

Security NONE

Lambda function

Region eu-west-1

Name ServerlessYammer

ApplicationName ServerlessYammer

Runtime python3.6

Blueprint empty-python

Handler lambda_function.lambda_handler

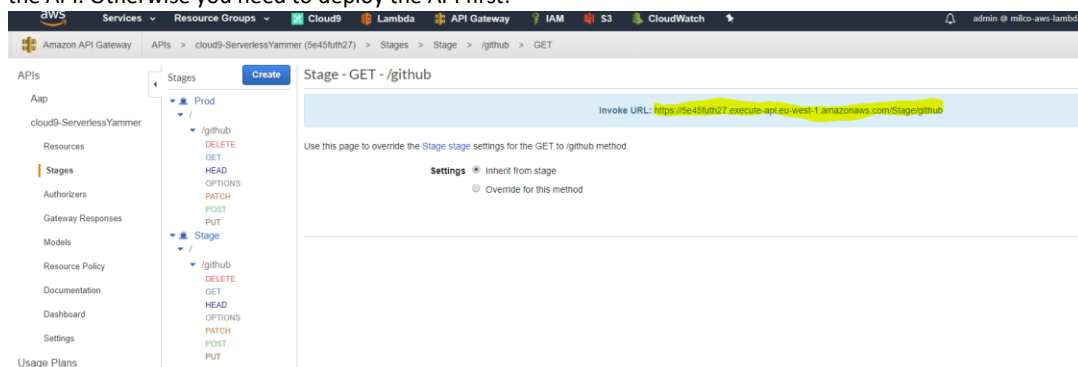
Memory-size 128

Role Automatically Generated

Region: eu-west-1

Previous Finish

- 14) Verify if the API has been deployed to a stage (API > Stages). If it has been deployed, pick one of the endpoints for the API. Otherwise you need to deploy the API first:



Navigate to Github, add a new webhook from the repository settings in the repository you created for this purpose:

The screenshot shows the GitHub 'Add webhook' interface. On the left is a sidebar with navigation links: Options, Collaborators, Branches, Webhooks (highlighted), Integrations & services, and Deploy keys. Below these are links for Moderation and Interaction limits. The main content area is titled 'Webhooks / Add webhook'. It contains the following sections:

- Introduction:** A paragraph explaining that GitHub will send a POST request to the specified URL with details of subscribed events. It also mentions that the data format can be JSON, x-www-form-urlencoded, etc., and refers to developer documentation.
- Payload URL:** A text input field containing the URL: `https://5e45futh27.execute-api.eu-west-1.amazonaws.com/Stage/git`.
- Content type:** A dropdown menu set to `application/json`.
- Secret:** An empty text input field for a secret key.
- SSL verification:** A section stating that by default, SSL certificates are verified. It has two radio buttons: **Enable SSL verification** (selected) and **Disable (not recommended)**.
- 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 labeled **Active**, with a note: 'We will deliver event details when this hook is triggered.'

At the bottom of the form is a green button labeled 'Add webhook'.

- 15) Now commit your code from Cloud9's terminal window and push this to Github. This should trigger the actual lambda function:
 (From my home directory)
`git add ServerlessYammer` # add all resources in my serverless application directory
`git commit -m "Initial commit, empty shell"`
`git push --set-upstream origin master`
- 16) Verify in CloudWatch that your lambda function has been triggered.

17) From Github you can also inspect the payload that has been sent and the response that Github has received:

☒ Just the push event.
 ☐ Send me everything.
 ☐ Let me select individual events.

☒ Active
 We will deliver event details when this hook is triggered.

Update webhook
 Delete webhook

Recent Deliveries

328d5392-ac4b-11e8-8726-3d39ef185975
 2018-08-30 13:52:40

Request
 Response 502
 Redeliver
 Completed in 0.14 seconds.

Headers

```

Request URL: https://5e45futh27.execute-api.eu-west-1.amazonaws.com/Stage/github
Request method: POST
content-type: application/json
Expect:
User-Agent: GitHub-Hookshot/ea6a0a6
X-GitHub-Delivery: 328d5392-ac4b-11e8-8726-3d39ef185975
X-GitHub-Event: push
      
```

Payload

```

{
  "ref": "refs/heads/master",
  "before": "172086f5f0cb15f5bf2547e4be34ae129f702b9a",
  "after": "4b93f928a86c259628c88f7f9c95dccc66e695ec",
  "created": false,
  "deleted": false,
  "forced": false,
  "base_ref": null,
  "compare": "https://github.com/mnuman/serverless-yammer/compare/172086f5f0cb...4b93f928a86c",
  "commits": [
    f
      
```

Currently, API Gateway still returns an error message (statuscode 5xx) as we did not put in any effort to format the answer properly – for now that is okay:

Recent Deliveries

328d5392-ac4b-11e8-8726-3d39ef185975
 2018-08-30 13:52:40

Request
 Response 502
 Redeliver
 Completed in 0.14 seconds.

Headers

```

Connection: keep-alive
Content-Length: 36
Content-Type: application/json
Date: Thu, 30 Aug 2018 11:52:40 GMT
Via: 1.1 15a3e53929b8b98c1afabel7cca4b1fd.cloudfront.net (CloudFront)
x-amz-apigw-id: Mb8VXhp6joeFzAA=
X-Amz-Cf-Id: yLHXhSvOaPlQlKG2AjYyZhrEvV-ckQSuZ9B0Jgk0KCYCKx3Q-ztwv==
x-amzn-RequestId: 32a726fb-ac4b-11e8-9ace-0d5f8b74a77d
X-Cache: Error from cloudfront
      
```

Body

```

{"message": "Internal server error"}
      
```

fbae1f59-ac4a-11e8-96e5-f92116c054dd
 2018-08-30 13:51:08

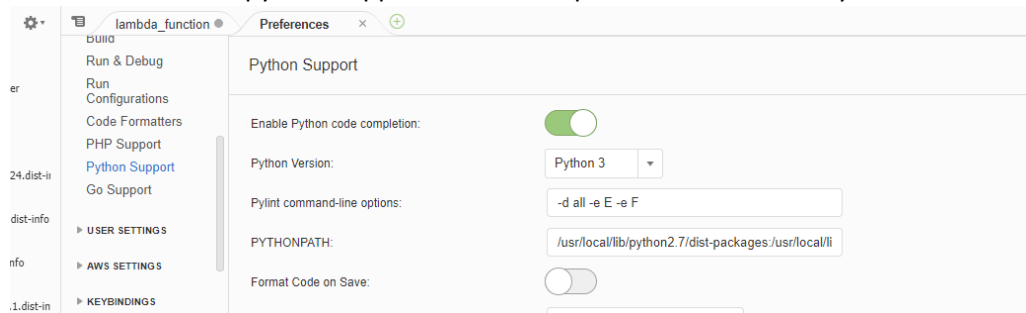
18) Change into your **Application's** directory using the Cloud9 terminal, enable the virtual environment by source the activate script from `venv/bin/activate` and install the requests package locally:

```

admin:~/environment (master) $ cd ServerlessYammer/
admin:~/environment/ServerlessYammer (master) $ source venv/bin/activate
(venv) admin:~/environment/ServerlessYammer (master) $
  
```

`pip install requests -t .`

- 19) Make sure that the python support in Cloud9's preferences is set to Python3:



- 20) Define some implementation code to compose a message (to be sent out later) and prepare a response for GitHub:

```
import json
import requests
def lambda_handler(event, context):
    body = json.loads(event['body'])
    myMessage = f"Hi, {body['head_commit']['author']['name']} has just
    committed code at {body['head_commit']['url']} with message
    {body['head_commit']['message']}"

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

- 21) Deploy your code to AWS

- 22) Again, commit your changes locally and commit your change to Github. Now verify that Github receives a proper response:

Recent Deliveries

✓

ed6c3bde-ac4f-11e8-9284-ee91d74c564a

2018-08-30 14:26:32

...

Request

Response 200

Redeliver

🕒 Completed in 0.6 seconds.

Headers

Connection: keep-alive

Content-Length: 241

Content-Type: application/json

Date: Thu, 30 Aug 2018 12:26:32 GMT

Via: 1.1 41d0ea27e141b983263dc5529dcc1ea5.cloudfront.net (CloudFront)

x-amz-apigw-id: McBS0E99DoEFETQ=

X-Amz-Cf-Id: 2Ow4ivNPhUwmiHzzq6eXqahetM4i50k0jADpeYY120nbu8IZH9kr-g==

x-amzn-RequestId: eda3f808-ac4f-11e8-a099-4b78be23626d

X-Amzn-Trace-Id: Root=1-5b87e278-d4fd4d982d99eb60d14dc378;Sampled=0

X-Cache: Miss from cloudfront

Body

"Hi, Milco Numan has just committed code at https://github.com/mnuman/serverless-yammer/commit/543ac

◀ ▶

- 23) Now we're ready to push the message to the world ...

- 24) In order to do this, you will need the Yammer developer token you jotted down earlier; we're leveraging the Yammer REST API (that's why you installed the requests package). This API is documented [here](#). To authenticate your call, you need to pass in a dictionary in the headers parameters with the HTTP headers; the

header's name must be "Authorization", it's value is your developer token prefixed by "Bearer " (that is Bearer followed by a single space (OAuth).

The address for the Yammer API is <https://www.yammer.com/api/v1/messages.json>. The payload is entered as a dictionary in the data parameter, this dictionary must have a body key holding the actual message and a group_id holding the id of the AWS-Yammer group:

```
import json
import requests
def lambda_handler(event, context):
    print(json.dumps(event))

    body = json.loads(event['body'])
    myMessage = f"Hi, {body['head_commit']['author']['name']} has just
committed code at {body['head_commit']['url']} with message
{body['head_commit']['message']}"
    auth = { "Authorization" : "Bearer 10691656-iLf4mZMqSZzQlnAKeQq4aA"}
    payload = { "body" : myMessage, "group_id" : 15767042}
    r = requests.post("https://www.yammer.com/api/v1/messages.json",
headers=auth, data=payload)

    print("Status code Yammer call:" + str(r.status_code))

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

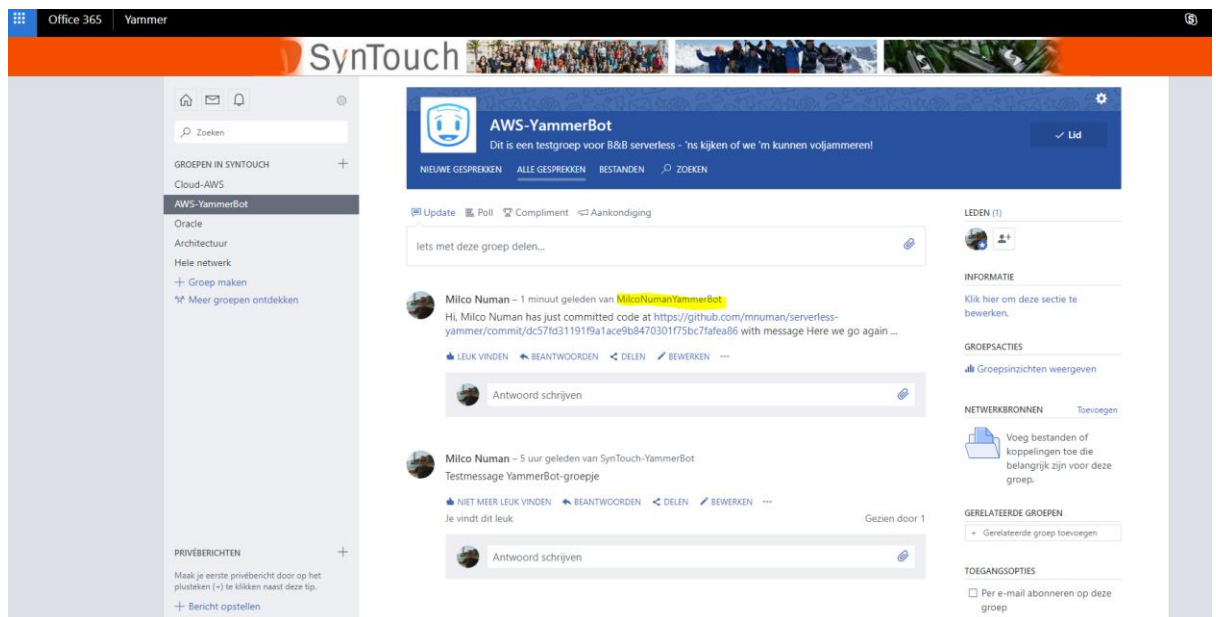
Should you mess up (e.g. commit and push the code and forget to redeploy it first ...), you can also redeliver message from GitHub:

Recent Deliveries

The screenshot shows the 'Recent Deliveries' section of a GitHub repository. At the top, there is a delivery ID 'bd46c38e-ac54-11e8-8fe7-087c79a1cf13' with a red warning icon and a timestamp '2018-08-30 15:00:58'. Below this, there are tabs for 'Request' and 'Response', with 'Response' selected and showing a status of '502'. A 'Redeliver' button is visible. The 'Headers' section is expanded, showing the following details:

```
Request URL: https://5e45futh27.execute-api.eu-west-1.amazonaws.com/Stage/github
Request method: POST
content-type: application/json
Expect:
User-Agent: GitHub-Hookshot/ea6a0a6
X-GitHub-Delivery: bd46c38e-ac54-11e8-8fe7-087c79a1cf13
X-GitHub-Event: push
```

And then, finally ...



As an extra exercise, retrieve the authentication token from the function's environment variables (or from the secrets manager) – e.g. us [os.getenv](#) to retrieve values.



AWS-YammerBot

Dit is een testgroep voor B&B serverless - 'ns kijken of we 'm kunnen voljammeren!

NIEUWE GESPREKKEN

ALLE GESPREKKEN

BESTANDEN

ZOEKEN

Update Poll Compliment Aankondiging

Iets met deze groep delen...



Milco Numan – 50 seconden geleden van MilcoNumanYammerBot

Hi, Milco Numan has just committed code at <https://github.com/mnuman/serverless-yammer/commit/9d5234ab9d8e23c01fd7bbd06214a1e608ff4992> with message Invalidated my developer token and generated a new one; pushed the token into a environment variable set MANUALLY on the Lambda function after deployment

LEUK VINDEN BEANTWOORDEN DELEN BEWERKEN ...



Antwoord schrijven



Milco Numan – 14 minuten geleden van MilcoNumanYammerBot

Hi, Milco Numan has just committed code at <https://github.com/mnuman/serverless-yammer/commit/dc57fd31191f9a1ace9b8470301f75bc7fafea86> with message Here we go again ...

LEUK VINDEN BEANTWOORDEN DELEN BEWERKEN ...



Antwoord schrijven

