

퇴근봇 채용하기

@shia

퇴근봇 채용하기

라고 쓰고 만든다고 읽습니다

@shia


옛날 옛적에
어떤 개발자가 살고 있었어요.

그 개발자는 칼퇴근에 익숙하지 않은,
숨쉬듯 야근하는 개발자였습니다.

퇴근 시간을 알려주는 봇이 없다면,
언제까지고 일할 것만 같았죠.



kitakubot BOT 6:25 PM

MINASWAN  5分前です。コミットしましょう。

MINASWAN  帰りましょう。

그러던 어느날,
퇴근봇이 돌연 사표를 냈어요.

이 이야기는 퇴근뭇을 빨리 채용하고
싶었던 한 개발자의 이야기입니다.

제 공
이상한 모임

이 발표에 등장하는 고유명사는
실제 고유명사와 관련이 많습니다.

방을 밝게 하고 화면으로부터
50cm 이상 떨어져서 시청해주세요.

아무튼 그래서 급하게
만들어본 채용공고

채용 공고

- 포지션: 퇴근 시간 안내 담당(6:25, 6:30)
- 근무 시간: 주 5일(월-금) 근무
- 급여: 상담 후 결정

채용 공고

- 포지션: 퇴근 시간 안내 담당(6:25, 6:30)
- 근무 시간: 주 5일(월-금) 근무
- 급여: 상담 후 결정
- 우대사항:
 - 개발자들의 건강에 지대한 관심이 있으신 분
 - 단순 작업을 좋아하시는 분
 - 열정 페이가 가능하신 분

그렇게 받은 채용 공고

이력서1

- 이름: Cron
- 희망 연봉: \$14 * 12 (EC2 t1.micro)
- 자기소개
 - 단순 작업 전문입니다.

이력서1

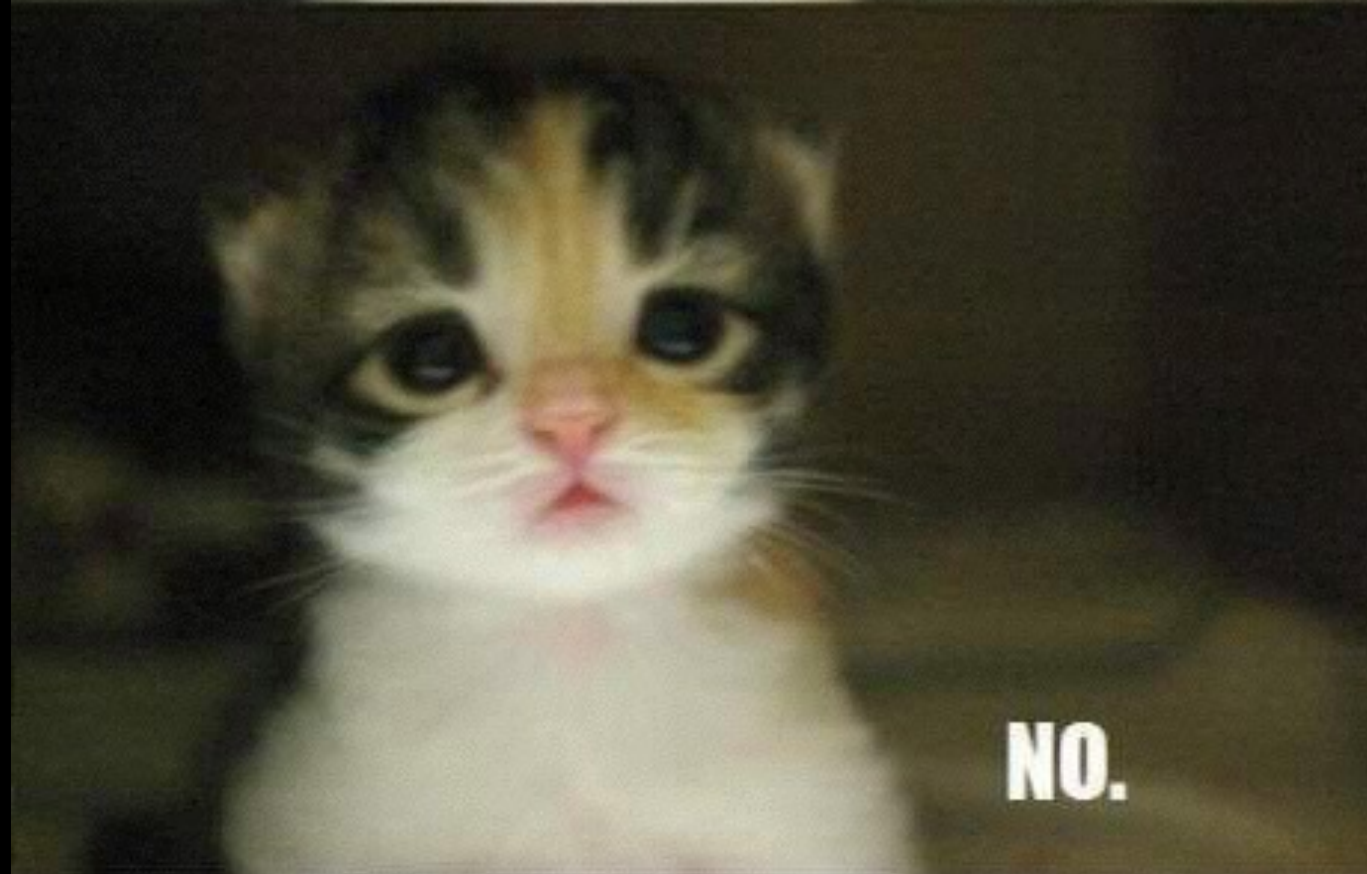
- 이름: Cron
- 희망 연봉: \$14 * 12 (EC2 t1.micro)
- 자기소개
 - 단순 작업 전문입니다.

이력서1

- 희망 연봉: \$14 * 12 (EC2 t1.micro)



HAHAHA



NO.

다음

이력서2

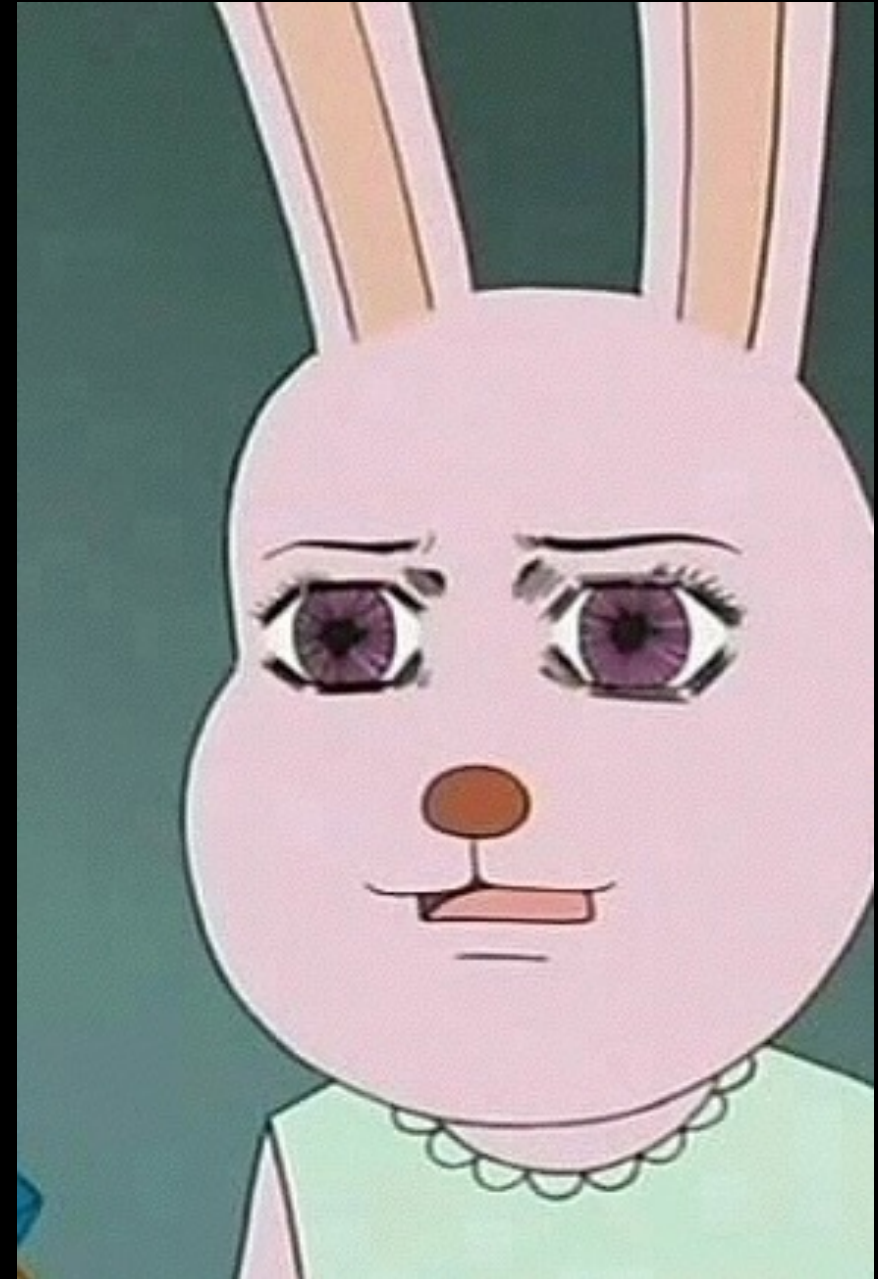
- 이름: Hubot
- 특이사항
 - 이미 일하는 중
 - 추가 업무 희망
 - 연봉 인상 : \$0

이력서2

- 이름: Hubot
- 특이사항
 - 이미 일하는 중
 - 추가 업무 희망
 - 연봉 인상 : \$0

이력서2

- 이름: Hubot
- 특이사항
 - 이미 일하는 중
 - 추가 업무 희망
 - 연봉 인상 : \$0



일단 한번 시켜보기로 함

node-cron

```
npm install cron --save
```

```
cron = require("cron").CronJob
```

```
module.exports = (robot) ->
```

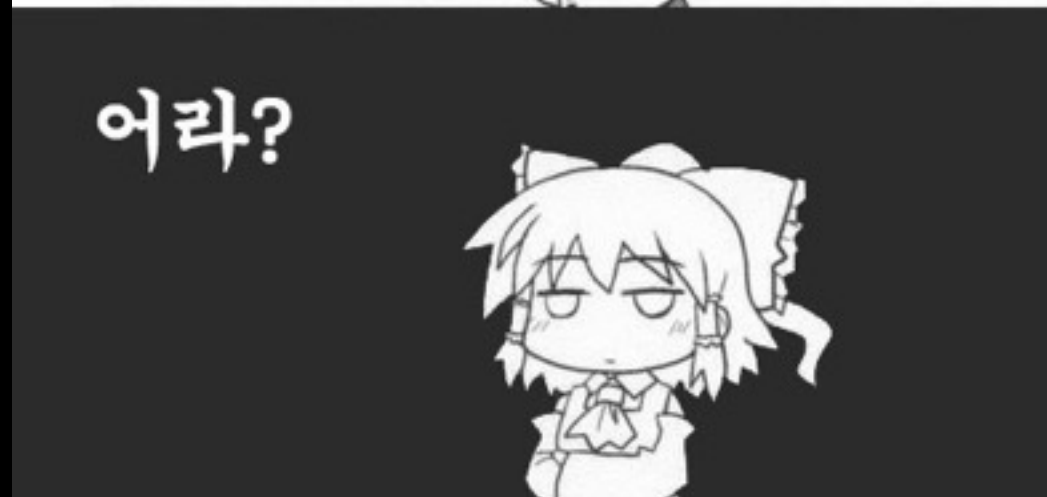
```
  new cron "0 0 18 30 * 1-5", () =>
```

```
    robot.send {room: "#general"}, "帰りましょう。"
```

```
  , null, true
```



어라?





개발 환경에서는 잘 되는데,
정작 Heroku에서는 동작을 안함

원인을 알 수가 없어서 디버깅은
해야겠는데, 그 시간이 너무 아까움

그런 이유로 다른 대안을
찾아보기로 했습니다.

그런 이유로 다른 대안을
찾아보기로 했습니다.

~~그거 찾을 시간에 다버깅했으면 해결했을텐데...~~

그리고 얼마 뒤, 받은 제보
(Thanks to @originerd)

"AWS Lambda에서
Schedule job 돌릴 수 있어요"



이력서3

- 이름: Serverless
- 희망 연봉: \$0 * 12
- 특이사항
 - 혼자서는 일을 못함

이력서3

- 이름: Serverless
- 희망 연봉: \$0 * 12
- 특이사항
 - 혼자서는 일을 못함

이력서3

- 이름: Serverless
- 희망 연봉: \$0 * 12
- 특이사항
 - 혼자서는 일을 못함

어떻게 일하실 거죠?

"CloudWatch Event에서
Slack incoming hook을 호출하는
Lambda 함수를 실행할 겁니다."

...?

CloudWatch?
Lambda?



CloudWatch Event

Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in Amazon Web Services (AWS) resources to AWS Lambda functions, Amazon SNS topics, Amazon SQS queues, streams in Amazon Kinesis Streams, or built-in targets.

CloudWatch Event

Amazon CloudWatch Events **delivers** a near real-time stream of **system events** that describe changes in Amazon Web Services (AWS) resources **to AWS Lambda functions**, Amazon SNS topics, Amazon SQS queues, streams in Amazon Kinesis Streams, or built-in targets.

Lambda Function

AWS Lambda is a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you.

Lambda Function

AWS Lambda is a **serverless** compute service that **runs your code in response to events** and automatically manages the underlying compute resources for you.

CloudWatch Event

- Approximately \$1.00 per million custom events generated

Lambda

- Depend on memory & time & traffic usage

1 Month

Memory: 128M

Time: 7.5 sec

Traffic: Less than 1MB

-> 실질 무료

5-min bot cooking

1. 웹훅을 준비한다

1. 웹훅을 준비한다
2. 트리거를 설정한다.

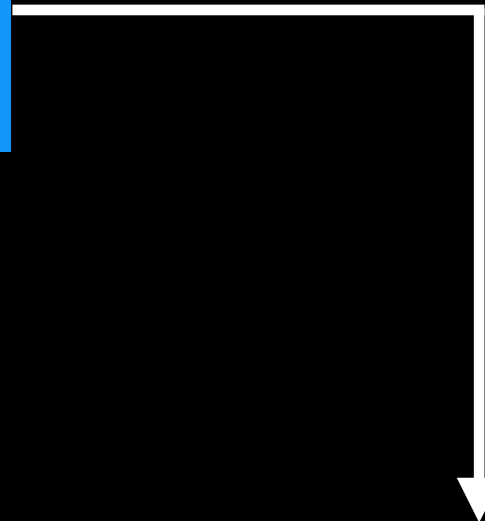
1. 웹훅을 준비한다
2. 트리거를 설정한다.
3. 람다 함수를 만든다

그림 한장으로 살펴보는
작업 흐름

CloudWatch

설정된 시각에
람다를 호출

Lambda

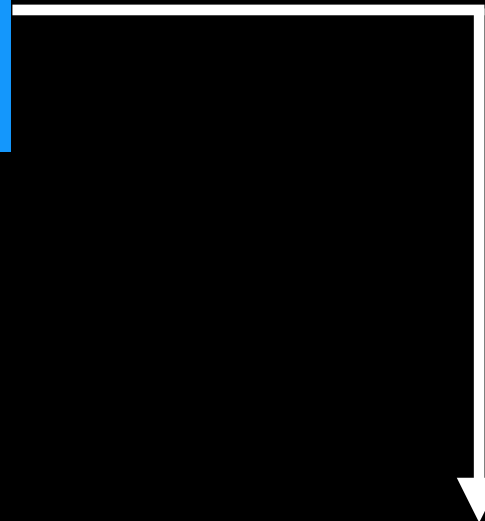


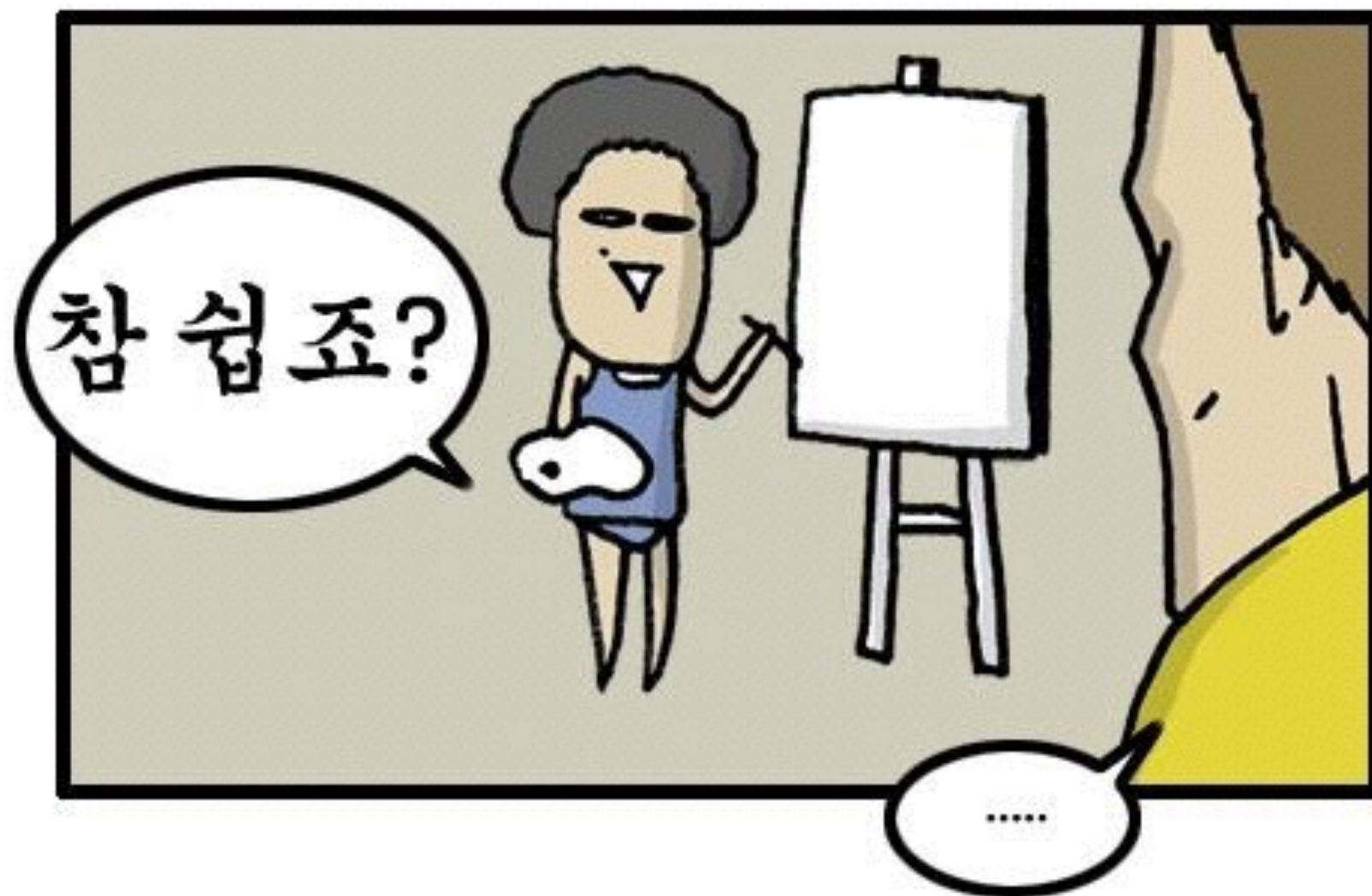
CloudWatch

Lambda

각 람다는 슬랙 웹훅을 호출

Slack





1. 웹훅을 준비하기



[Add Configuration](#)

[Help and support](#) >

[Privacy policy](#) >

Incoming WebHooks

Incoming Webhooks are a simple way to post data to a server. They make use of normal HTTP requests with a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhook messages that stand out from regular chat messages.



Weirdmeetup

Only select team members can install the app

2. 트리거를 설정하기

Configure triggers

Configure an optional trigger to automatically invoke your function.



Lambda

Remove

Cancel

Previous

Next

Configure triggers

Configure an optional trigger to automatically invoke your function.



Lambda

Remove

Cancel

Previous

Next

Configure triggers

Configure an optional trigger to automatically invoke your function.



Lambda

Remove

Filter integrations



API Gateway



AWS IoT



CloudWatch Events - Schedule



CloudWatch Logs



Cognito Sync Trigger



DynamoDB



Kinesis



S3

Cancel

Previous

Next

Configure triggers

Configure an optional trigger to automatically invoke your function.



Lambda

Remove

Filter integrations



API Gateway



AWS IoT



CloudWatch Events - Schedule



CloudWatch Logs



Cognito Sync Trigger



DynamoDB



Kinesis



S3

Cancel

Previous

Next

Configure triggers

Configure an optional trigger to automatically invoke your function.

CloudWatch Events - Schedule



Lambda

Remove

Rule name

before-5-min



Rule description

trigger incomming hook to #_general



Schedule expression

cron(25 9 ? * MON-FRI *)



Lambda will add the necessary permissions for CloudWatch Events to invoke your Lambda function on a schedule. [Learn more](#) about the Lambda permissions model.

Enable trigger



Cancel

Previous

Next

Configure triggers

Configure an optional trigger to automatically invoke your function.

CloudWatch Events - Schedule



Lambda

Remove

Rule name

before-5-min



Rule description

trigger incomming hook to #_general



Schedule expression

cron(25 9 ? * MON-FRI *)



UTC Time

Lambda will add the necessary permissions for CloudWatch Events to invoke your Lambda function on a schedule. [Learn more](#) about the Lambda permissions model.

Enable trigger



Cancel

Previous

Next

Configure triggers

Configure an optional trigger to automatically invoke your function.

CloudWatch Events - Schedule



Lambda

Remove

Rule name

before-5-min



Rule description

trigger incomming hook to #_general



Schedule expression

cron(25 9 ? * MON-FRI *)



UTC Time

Lambda will add the necessary permissions for CloudWatch Events to invoke your Lambda function on a schedule. [Learn more](#) about the Lambda permissions model.

Enable trigger



Cancel

Previous

Next

3. 람다 함수 설정하기

Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Name*

Description

Runtime*



Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Name*

Description

Runtime*

NodeJS ▼

Node 4.3 Recommended

Code entry type

Edit code inline



```
1  const exec = require("child_process").exec;
i 2  const message = "퇴근 5분 전입니다. 커밋해주세요. :octocat:"
i 3  const payload = `payload={"channel": "#_general", "username": "퇴근
i 4  const url = "https://hooks.slack.com/services/
5
6  exports.handler = (_event, context) => {
i 7      const cmd = `curl -X POST --data-urlencode '${payload}' ${url}`
8
9      child = exec(cmd, (error) => {
10          // Resolve with result of process
i 11          context.done(error, "Incomming Hook is triggered!")
i 12      })
13
14      // Log process stdout and stderr
i 15      child.stdout.on("data", console.log)
i 16      child.stderr.on("data", console.error)
```

```
const exec = require("child_process").exec;
const message = "퇴근 5분 전입니다. 커밋해주세요. :octocat:"
const payload = \
  `payload={"channel": "#_general", "username": "퇴근봇",
            "text": "${message}", "icon_emoji": ":weirdmeetup:"}`
const url = "https://hooks.slack.com/services/...."
```



```
exports.handler = (_event, context) => {  
  const cmd = `curl -X POST --data-urlencode '${payload}' ${url}`  
  
  child = exec(cmd, (error) => {  
    // Resolve with result of process  
    context.done(error, "Incomming Hook is triggered!")  
  })  
  
  // Log process stdout and stderr  
  child.stdout.on("data", console.log)  
  child.stderr.on("data", console.error)  
}
```

```
exports.handler = (_event, context) => {  
  const cmd = `curl -X POST --data-urlencode '${payload}' ${url}`  
  
  child = exec(cmd, (error) => {  
    // Resolve with result of process  
    context.done(error, "Incomming Hook is triggered!")  
  })  
  
  // Log process stdout and stderr  
  child.stdout.on("data", console.log)  
  child.stderr.on("data", console.error)  
}
```

```
exports.handler = (_event, context) => {  
  const cmd = `curl -X POST --data-urlencode '${payload}' ${url}`  
  
  child = exec(cmd, (error) => {  
    // Resolve with result of process  
    context.done(error, "Incomming Hook is triggered!")  
  })  
  
  // Log process stdout and stderr  
  child.stdout.on("data", console.log)  
  child.stderr.on("data", console.error)  
}
```

Lambda function handler and role

Handler*

index.handler

Role*

Create new role from template(s) ▼



Lambda will automatically create a role with permissions from the selected policy templates. Note that basic Lambda permissions (logging to CloudWatch) will automatically be added. If your function accesses a VPC, VPC permissions will also be added.

Role name*

leave-work-role



Policy templates

✱ S3 object read-only perm...



Advanced settings

These settings allow you to control the code execution performance and costs for your Lambda function. Changing your resource settings (by selecting memory) or changing the timeout may impact your function cost. [Learn more](#) about how Lambda pricing works.

Memory (MB)*

128



Timeout*

0



min

3



sec

All AWS Lambda functions run securely inside a default system-managed VPC. However, you can optionally configure Lambda to access resources, such as databases, within your custom VPC. [Learn more](#) about accessing VPCs within Lambda. **Please ensure your role has appropriate permissions to configure VPC.**

VPC

No VPC



동작 확인하기

Lambda > Functions > ready-to-commit

Qualifiers ▼

Test

Actions ▼

Code

Configuration

Triggers

Monitoring

Code entry type

Edit code inline ▼

```
1  const exec = require("child_process").exec;
i 2  const message = "퇴근 5분 전입니다. 커밋해주세요. :octocat:"
i 3  const payload = `payload={"channel": "#_general", "username": "퇴근
i 4  const url = "https://hooks.slack.com/services/
5
6  exports.handler = (_event, context) => {
i 7      const cmd = `curl -X POST --data-urlencode '${payload}' ${url}`
```

Lambda > Functions > ready-to-commit

Qualifiers ▼

Test

Actions ▼

Code

Configuration

Triggers

Monitoring

Code entry type

Edit code inline ▼

```
1  const exec = require("child_process").exec;
i 2  const message = "퇴근 5분 전입니다. 커밋해주세요. :octocat:"
i 3  const payload = `payload={"channel": "#_general", "username": "퇴근
i 4  const url = "https://hooks.slack.com/services/
5
6  exports.handler = (_event, context) => {
i 7      const cmd = `curl -X POST --data-urlencode '${payload}' ${url}`
```




이상한모임

퇴근봇 BOT 1:51 PM

퇴근 5분 전입니다. 커밋해주세요. 🐱

퇴근할 시간입니다. 오늘 하루도 수고 많으셨습니다. 🏃

정리하기

1. 세상은 넓고 스케줄 잡을 돌리는 법은 많다

1. 세상은 넓고 스케줄 잡을 돌리는 법은 많다
2. 간단한 & 저렴한 것을 원한다면,
CloudWatch & Lambda 조합도 괜찮다.

1. 세상은 넓고 스케줄 잡을 돌리는 법은 많다
2. 간단한 & 저렴한 것을 원한다면,
CloudWatch & Lambda 조합도 괜찮다.
3. Serverless는 생각보다 어렵지 않다.

그래서 개발자는 8시간 열심히 일하고
정시퇴근을 하며 행복하게 잘 살았습니다.

~ END ~

Any Question?



Thanks you!

