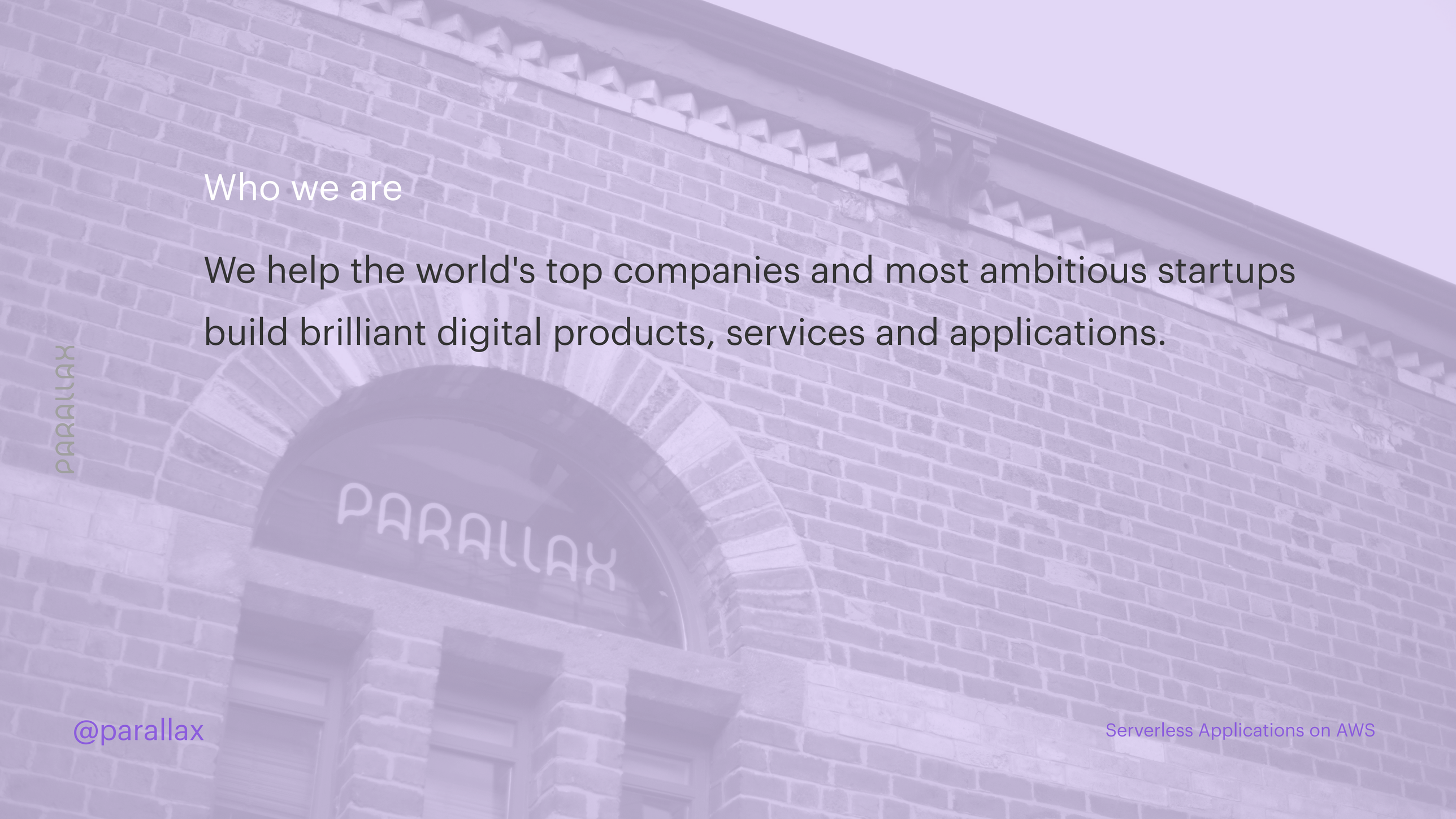
Scheduled Tasks Without a Server

James Hall, Director, Parallax
@MrRio

PARALLAX

@parallax



Who we are

We help the world's top companies and most ambitious startups build brilliant digital products, services and applications.

PARALLAX


Hisense

WALKER
MORRIS

cap hpi
FOR SMARTER AUTOMOTIVE DECISIONS™

NSF

AMERICAN
EXPRESS

ASDA

dyson

ARMY
BE THE BEST

BRITISH AIRWAYS

UEFA

amazon
web services™

enterprise
rent-a-car

Boston
Scientific

sky

NHS

Microsoft

ROCOL

sky SPORTS

THE SUNDAY TIMES

ARRIVA

Nestlé

Hermes
Makes delivery easy.

GoodHealth

CitiPark

adidas

Associated
British Foods
plc

easyGym

MP & SILVA

KERRY

SAP

PARALLAX

If you’re following along... run this now

PARALLAX

```
npm install -g serverless@alpha
```

Cron

PARALLAX

@parallax

What is cron?

Cron is a task scheduler that can run as often as every minute. It first shipped with Version 7 Unix in 1979.

What is it good for?

In web applications, examples include:

– Sending out subscription expiry emails

– Checking uptime

– Archiving files on S3

crontab

PARALLAX

@parallax

The Basics

```
crontab -e

* * * * * node /var/www/my/script.js

- - - - -
| | | | |
| | | | ----- Day of week (0 - 7) (Sunday=0 or 7)
| | | ------- Month (1 - 12)
| | ---------- Day of month (1 - 31)
| ------------ Hour (0 - 23)
-------------- Minute (0 - 59)
```

What's wrong with cron?

PARALLAX

@parallax

Modern Applications

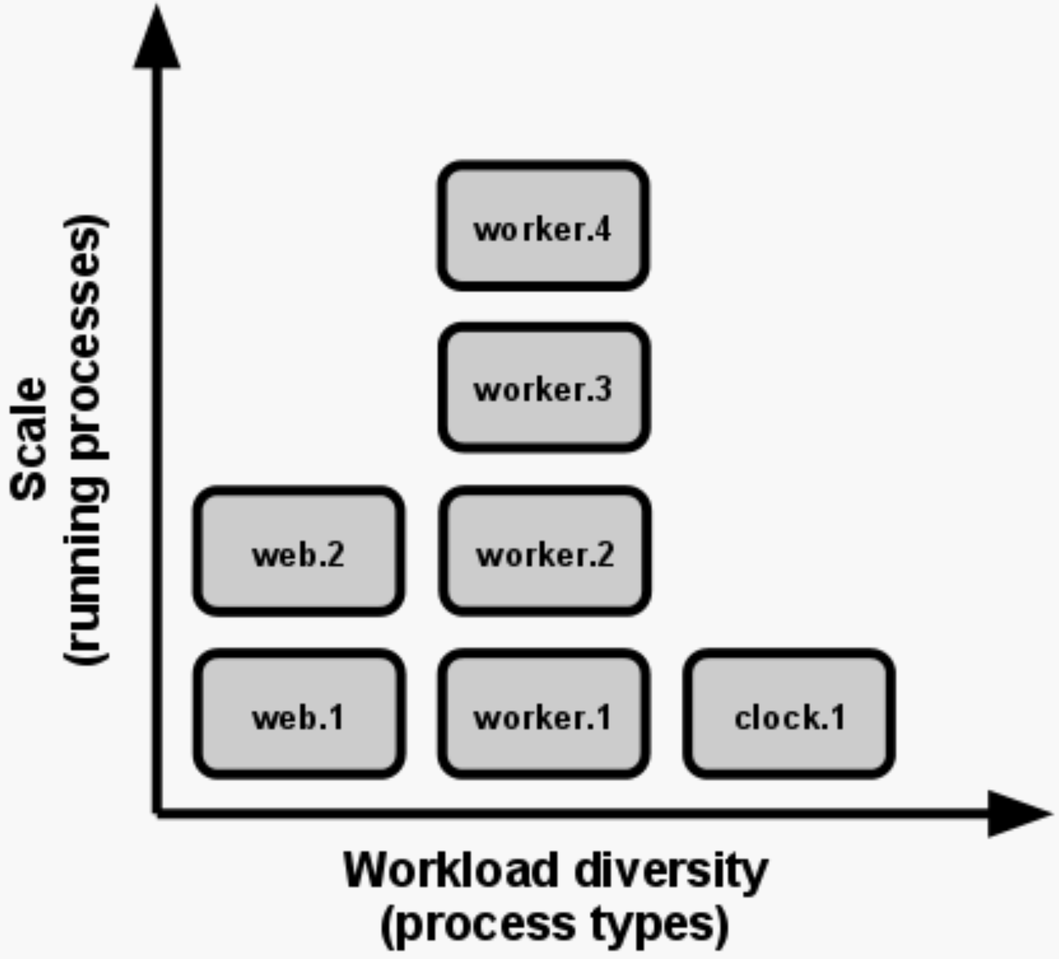
Modern web applications follow the Twelve Factors

(See: http://12factor.net/)

Rule 8 is **Concurrency**

Rule 9 is **Disposability**

Crontabs don’t work so well here



Serverless Framework

Installing

PARALLAX

```
npm install -g serverless@alpha
```

Setting AWS Credentials

Network

Elastic File System
Fully Managed File System for EC2

Glacier
Archive Storage in the Cloud

Snowball
Large Scale Data Transport

Storage Gateway
Hybrid Storage Integration

Database

RDS
Managed Relational Database Service

DynamoDB
Managed NoSQL Database

ElastiCache
In-Memory Cache

Redshift
Fast, Simple, Cost-Effective Data Warehousing

DMS

CloudTrail
Track User Activity and API Usage

Config
Track Resource Inventory and Changes

OpsWorks
Automate Operations with Chef

Service Catalog
Create and Use Standardized Products

Trusted Advisor
Optimize Performance and Security

Security & Identity

Identity & Access Management
Manage User Access and Encryption Keys

Directory Service
Host and Manage Active Directory

Inspector

Test Android, iOS, and Web Apps on Real Devices in the Cloud

Mobile Analytics
Collect, View and Export App Analytics

SNS
Push Notification Service

Application Services

API Gateway
Build, Deploy and Manage APIs

AppStream
Low Latency Application Streaming

CloudSearch
Managed Search Service

Elastic Transcoder
Easy-to-Use Scalable Media Transcoding

SES
Email Sending and Receiving Service

SQS

View your resources on the go with ou AWS Console mobile app, available from Amazon Appstore, Google Play, iTunes.

AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.

AWS re:Invent Announcements
Explore the next generation of AWS cloud capabilities. See what's new

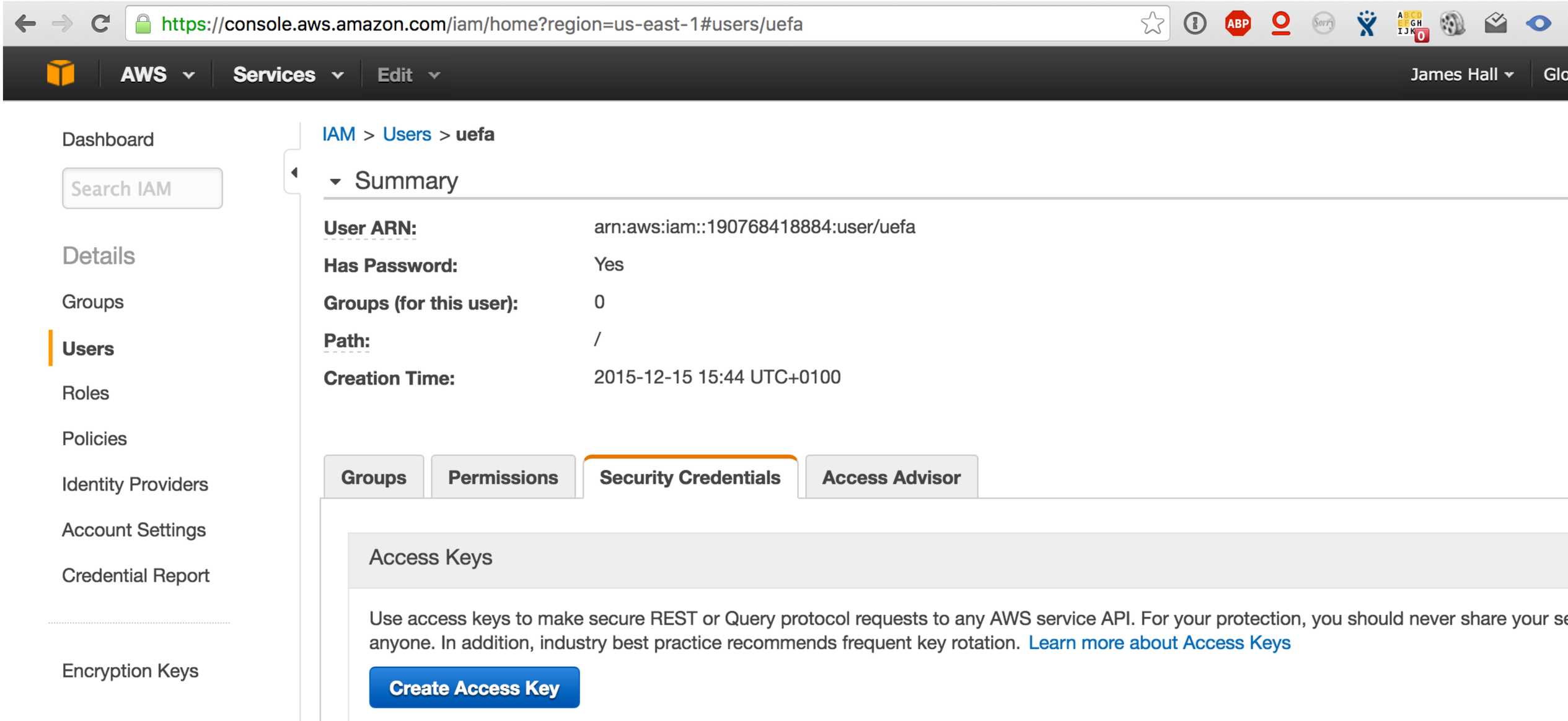
Service Health

All services operating normally.

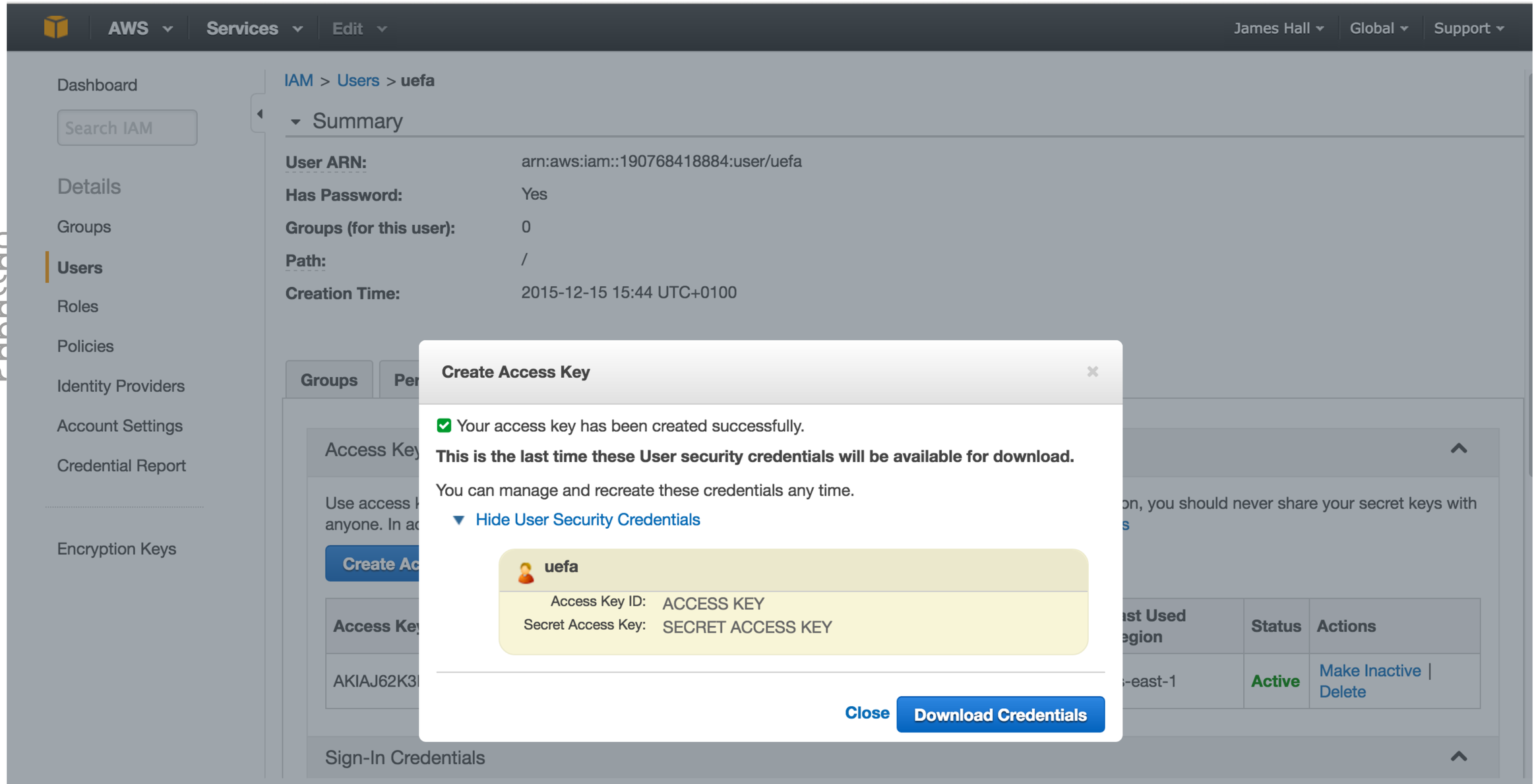
Updated: Jul 21 2016 17:00:00 GMT+0100

Service Health Dashboard

Setting AWS Credentials



Setting AWS Credentials



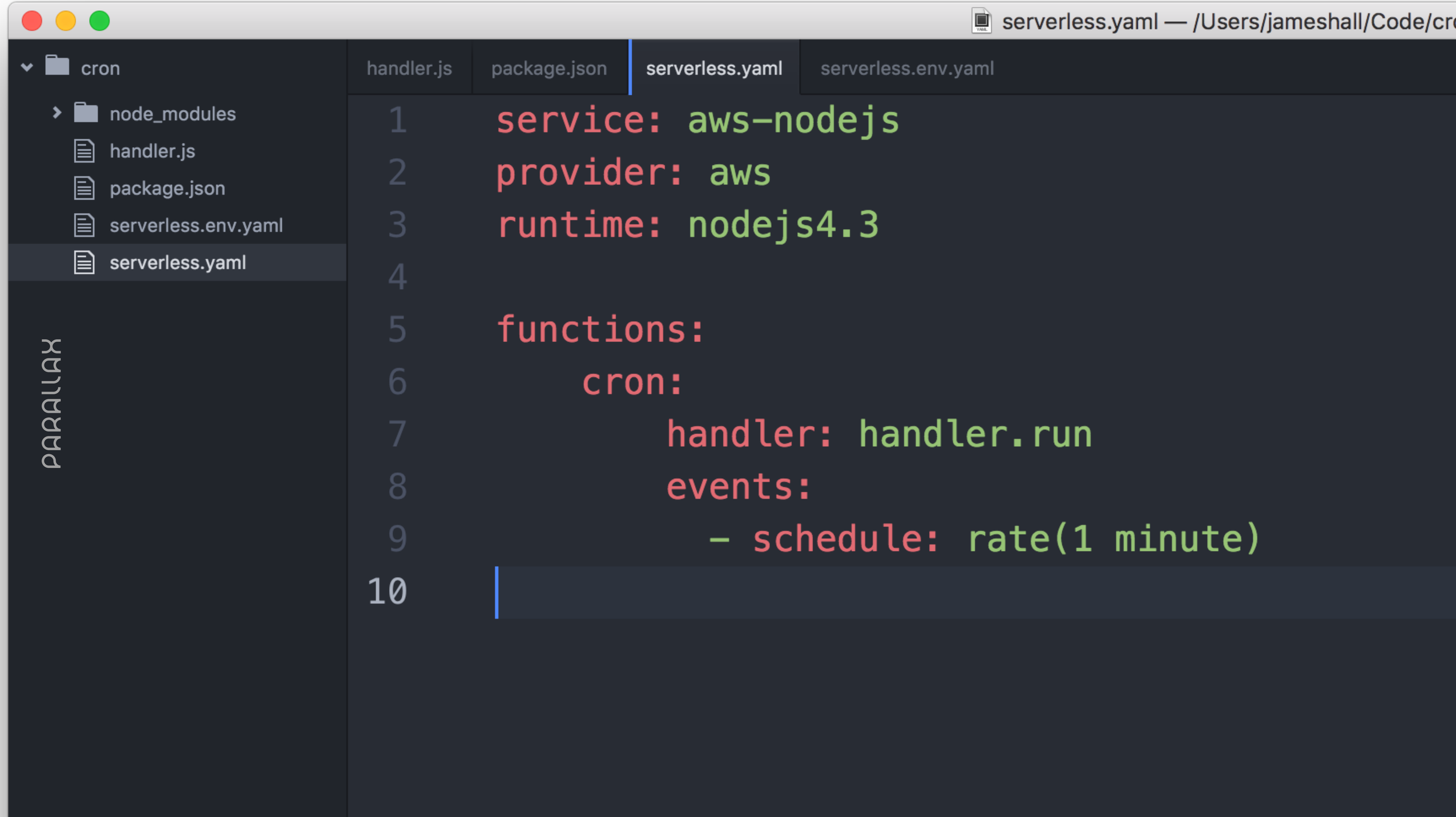
Setting AWS Credentials

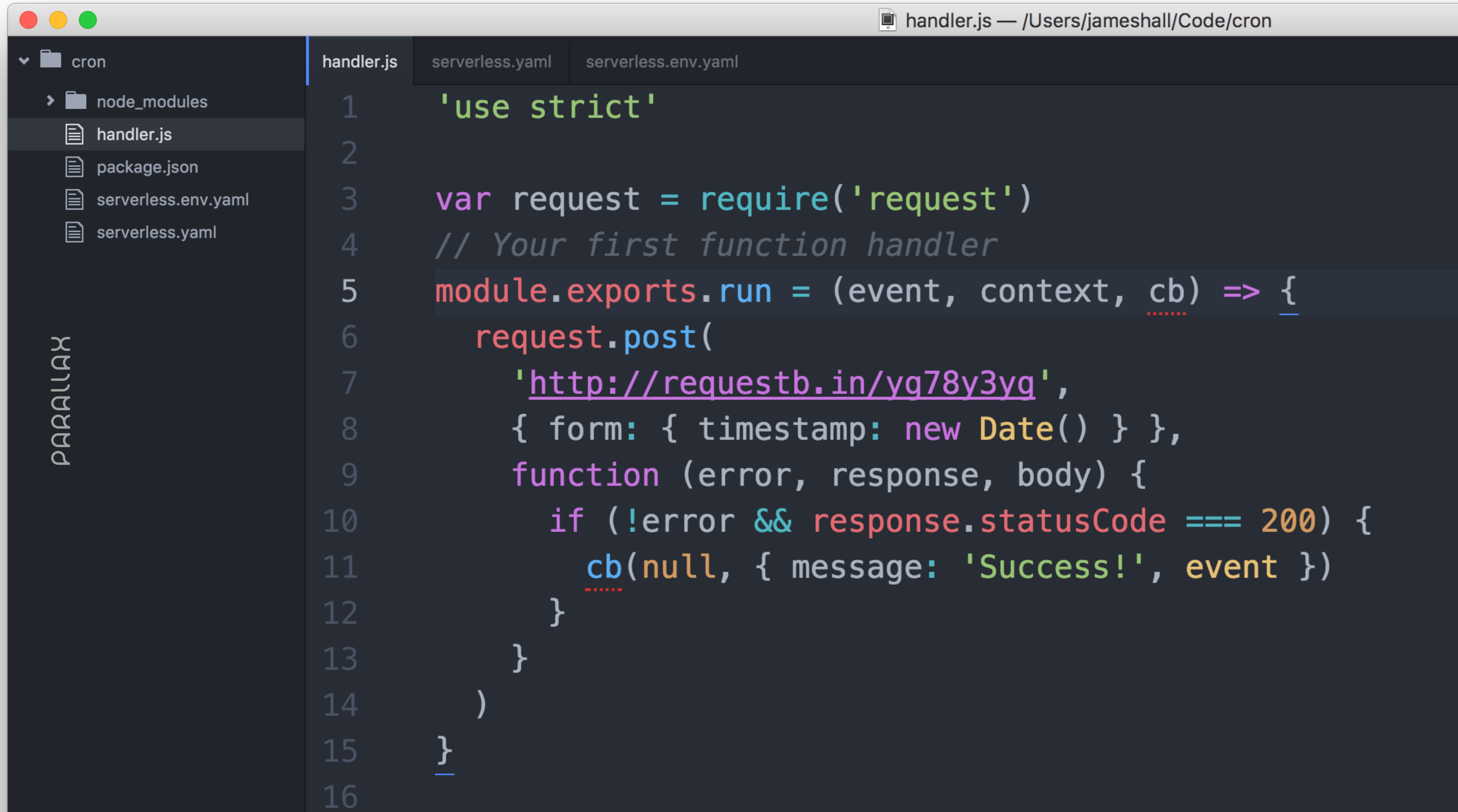
```
~/.aws/credentials
```

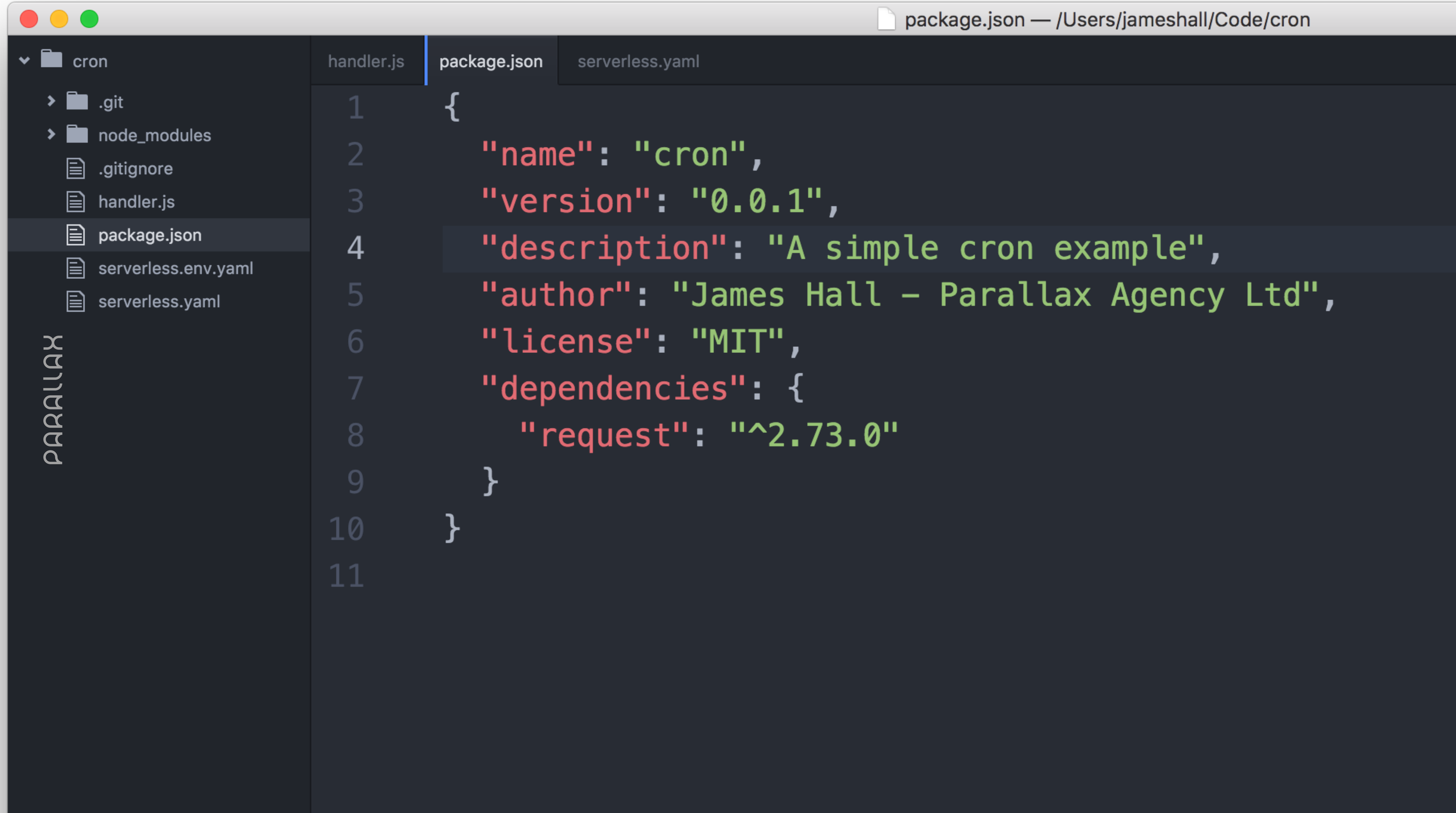
```
[default]
aws_access_key=THISISYOURACCESSKEY
aws_secret_access_key=ANDYOURSECRET1
```

Making a function

```
serverless create \
  --name cron \
  --provider aws \
  --template aws-nodejs
```


serverless.yaml — /Users/jameshall/Code/cr
cron
node_modules
handler.js
package.json
serverless.env.yaml
serverless.yaml
handler.js
package.json
serverless.env.yaml
service: aws-nodejs
provider: aws
runtime: nodejs4.3
functions:
cron:
handler: handler.run
events:
- schedule: rate(1 minute)
PARALLAX


handler.js — /Users/jameshall/Code/cron
cron
node_modules
handler.js
package.json
serverless.env.yaml
serverless.yaml
handler.js
serverless.yaml
serverless.env.yaml
'use strict'
var request = require('request')
// Your first function handler
module.exports.run = (event, context, cb) => {
request.post(
'http://requestb.in/yg78y3yg',
{ form: { timestamp: new Date() } },
function (error, response, body) {
if (!error && response.statusCode === 200) {
cb(null, { message: 'Success!', event })
}
}
)
}
PARALLAX


package.json — /Users/jameshall/Code/cron
cron
.git
node_modules
.gitignore
handler.js
package.json
serverless.env.yaml
serverless.yaml
handler.js
package.json
serverless.yaml
{
"name": "cron",
"version": "0.0.1",
"description": "A simple cron example",
"author": "James Hall - Parallax Agency Ltd",
"license": "MIT",
"dependencies": {
"request": "^2.73.0"
}
}
PARALLAX

```
Jamess-MacBook-Pro:cron jameshall$ serverless create --name cron --provider aws --template aws-nodejs
Serverless: Creating new Serverless service...
 _______                             __
|   _   .-----.----.--.--.-----.----|  .-----.-----.-----.
|   |___|  -__|   _|  |  |  -__|   _|  |  -__|__ --|__ --|
|____   |_____|__|  \___/|_____|__| |__|_____|_____|_____|
|   |   |              The Serverless Application Framework
|       |                           serverless.com, v1.0.0-alpha.2
 -------'

Serverless: Successfully created service in the current directory
Serverless: with template: "aws-nodejs"
Jamess-MacBook-Pro:cron jameshall$ sls deploy
Serverless: Creating Stack...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Checking stack creation progress...
Serverless: Stack successfully created.
Serverless: Zipping service...
Serverless: Uploading .zip file to S3...
Serverless: Updating Stack...
Serverless: Checking stack update progress...
Serverless: Deployment successful!
```

A Runscope Community Project — Learn more.

RequestBin

http://requestb.in/yg78y3yg

http://requestb.in
POST /yg78y3yg

</> application/x-www-form-urlencoded
38 bytes

9s ago
From 52.201.231.235, 162.158.76.143

FORM/POST PARAMETERS

timestamp: 2016-07-21T14:07:32.579Z

HEADERS

Cf-Ray: 2c5f3844b31c0f39-IAD
Content-Type: application/x-www-form-urlencoded
Connection: close
Total-Route-Time: 0
Accept-Encoding: gzip
Cf-Visitor: {"scheme":"http"}
X-Request-Id: 9dd9264a-8093-435c-a72e-ac6c9c45cca5
Via: 1.1 vegur
Content-Length: 38
Connect-Time: 1
Cf-Ipcountry: US
Host: requestb.in
Cf-Connecting-Ip: 52.201.231.235

RAW BODY

```
timestamp=2016-07-21T14%3A07%3A32.579Z
```

http://requestb.in
POST /yg78y3yg

</> application/x-www-form-urlencoded
38 bytes

1m ago
From 52.201.231.235, 173.245.56.95

PARALLAX

PARALLAX

Awesome. How does that work?

@parallax

CloudWatch
Dashboards
Alarms
ALARM 0
INSUFFICIENT 0
OK 0
Billing
Events
Rules
Logs
Metrics
Selected Metrics
Billing
CloudFront
DynamoDB
Events
Lambda
Logs
S3

Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

Create rule | Actions

Status: All | Name:

Viewing 1 to 1 of 1 Rules

	Status	Name	Description
○	✓	aws-nodejs-dev-helloScheduleEvent0-3MPISIX9XG40	

CloudWatch
Dashboards
Alarms
ALARM 0
INSUFFICIENT 0
OK 0
Billing
Events
Rules
Logs
Metrics
Selected Metrics
Billing
CloudFront
DynamoDB
Events
Lambda
Logs
S3

Rules > aws-nodejs-dev-helloScheduleEvent0-3MPISIX9XG40

Actions

Summary

ARN arn:aws:events:us-east-1:190768418884:rule/aws-nodejs-dev-helloScheduleEvent0-3MPISIX9XG40

Schedule Fixed rate of 1 minutes

Status Enabled

Description

Role ARN

Monitoring Show metrics of rules

Targets

Filter:

Viewing 1 to 1 of 1 Targets

Type	Resource name	Input
Lambda function	aws-nodejs-dev-hello	**Matched event**

CloudWatch Events - Schedule: aws-nodejs-dev-helloScheduleEvent0-3MPISIX9XG40

Disable | Delete

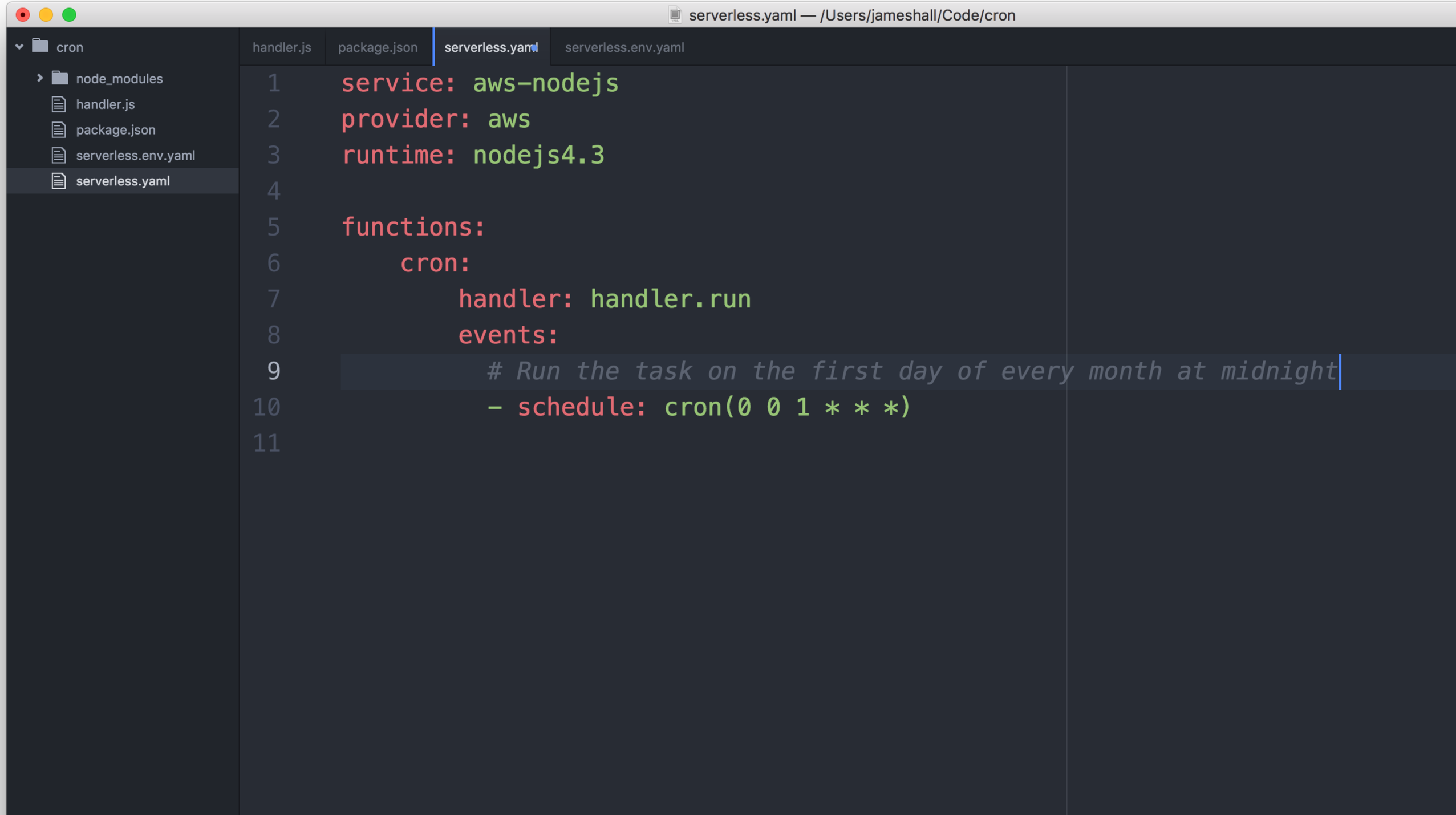
arn:aws:events:us-east-1:190768418884:rule/aws-nodejs-dev-helloScheduleEvent0-3MPISIX9XG40

Schedule expression: **rate(1 minute)** Description:

Add trigger

What about more complicated scheduling rules?

PARALLAX


serverless.yaml — /Users/jameshall/Code/cron
cron
node_modules
handler.js
package.json
serverless.env.yaml
serverless.yaml
handler.js
package.json
serverless.yaml
serverless.env.yaml
service: aws-nodejs
provider: aws
runtime: nodejs4.3
functions:
cron:
handler: handler.run
events:
Run the task on the first day of every month at midnight
- schedule: cron(0 0 1 * * *)

Slides and code:

https://github.com/parallax/serverless-cron-example

Thanks, any questions?

James Hall, Director, Parallax

@parallax
@MrRio

PARALLAX