

DAVID NOVICKI

---

# LEARNING SERVERLESS WORKSHOP

# WHO AM I?

- ▶ VP of Engineering at Dermveda
- ▶ Entrepreneur
- ▶ Sacramento Startup Developers
- ▶ Interests: Anything Javascript && Deep Learning



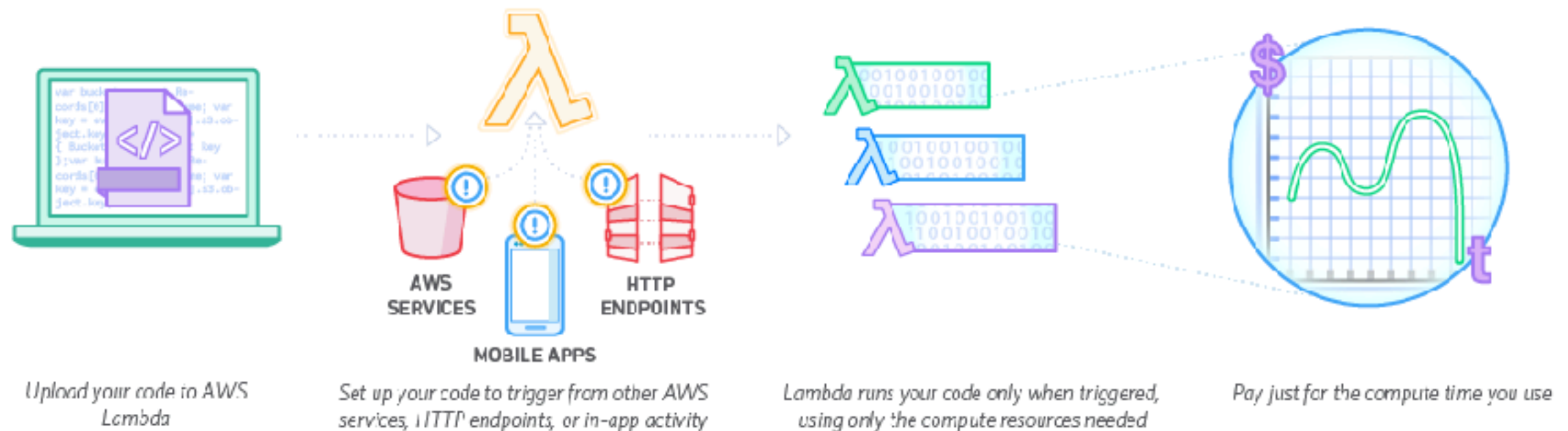
# OBJECTIVES

- ▶ Learn a 'code first' approach
- ▶ Gain an understanding of how Serverless works
- ▶ Deploy a serverless application
- ▶ Iterate application to build and save to a DynamoDB

# WHAT IS SERVERLESS?!

## JUST FUNCTIONS

### How It Works



# WHAT CLOUD PROVIDERS SUPPORT SERVERLESS?



[AWS Docs](#)



Microsoft Azure

[Azure Functions Docs](#)



IBM OpenWhisk

[OpenWhisk Docs](#)



Google Cloud Platform

[Cloud Functions Docs](#)



Kubeless

[Kubeless Docs](#)



spotinst

[Spotinst Docs](#)



webtasks

[Auth0 Webtasks Docs](#)

# WHY USE SERVERLESS?!

Serverless applications provide four main benefits:



### No server management

There is no need to provision or maintain any servers. There is no software or runtime to install, maintain, or administer.



### Flexible scaling

Your application can be scaled automatically or by adjusting its capacity through toggling the units of consumption (e.g. throughput, memory) rather than units of individual servers.



### High availability

Serverless applications have built-in availability and fault tolerance. You don't need to architect for these capabilities since the services running the application provide them by default.



### No idle capacity

You don't have to pay for idle capacity. There is no need to pre- or over-provision capacity for things like compute and storage. For example, there is no charge when your code is not running.

# *SERVER* *LESS*

```
# Install serverless globally
$ npm install serverless -g

# Login to your Serverless account
$ serverless login

# Create a serverless function
$ serverless create --template hello-world

# Deploy to cloud provider
$ serverless deploy

# Function deployed! Trigger with live url
$ http://xyz.amazonaws.com/hello-world
```



## AN ORGANIZATIONAL & DEPLOYMENT TOOL

serverless.yml

```
1  service: serverless-example
2
3  provider:
4    name: aws
5    runtime: nodejs6.10
6    stage: beta
7    region: us-west-2
8
9  functions:
10   hello:
11     handler: handler.hello
12     description: optional description for your Lambda
```

handler.js

```
3
4  module.exports.hello = (event, context, callback) => {
5    //implement code here
6  }
7
```

## CREATE AWS IAM KEY

The screenshot displays the AWS IAM console interface. At the top, the navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information (David, N. Virginia, Support). The main content area is divided into two sections. The left section, titled 'AWS services', features a search bar with 'IAM' entered, showing a dropdown list with 'IAM' and 'Manage User Access and Encryption Keys'. Below this, a grid of service icons is visible, including IAM, Lambda, EC2, Route 53, and S3. The right section, titled 'Helpful tips', contains two cards: 'Manage your costs' and 'Create an organization'. The bottom section of the console shows the 'Users' page. It includes a search bar with the text 'Find users by username or access key' and a 'Showing 1 result' indicator. A table lists the user 'serverless-admin' with the following details:

<input type="checkbox"/>	User name	Groups	Access key age	Password age	Last activity	MFA
<input type="checkbox"/>	serverless-admin	admin	⚠ 148 days	None	37 days	Not enabled

The left sidebar contains a navigation menu with links to Dashboard, Groups, Users, Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys.

## SET UP AWS CREDENTIALS

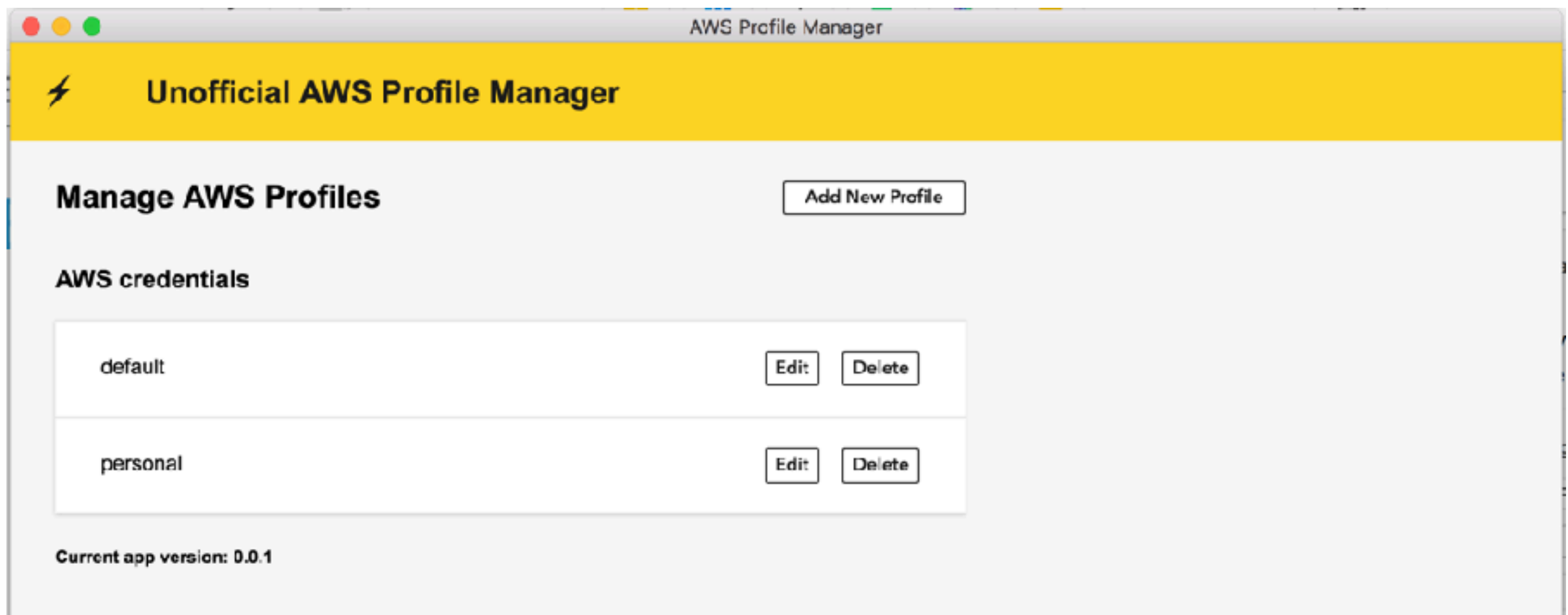
```
.aws — -bash — 90x50
[david-dmn: ~] ⚡ cd .aws && vi credentials
david-dmn: .aws ⚡
```

```
[default]
aws_access_key_id = [REDACTED]
aws_secret_access_key = [REDACTED]

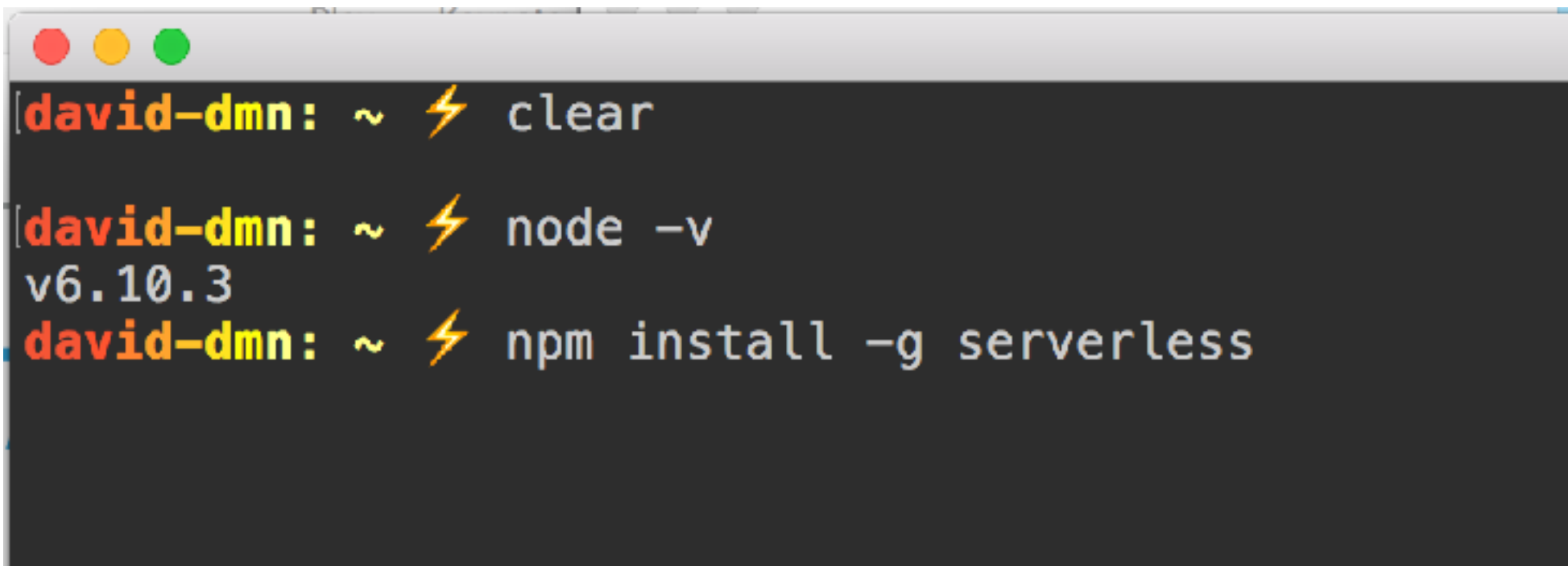
[personal]
aws_access_key_id = [REDACTED]
aws_secret_access_key : [REDACTED]
~
~
```

# AWS PROFILE MANAGER (UNOFFICIAL)

[HTTPS://GITHUB.COM/DAVIDWELLS/AWS-PROFILE-MANAGER](https://github.com/DAVIDWELLS/AWS-PROFILE-MANAGER)



## INSTALL SERVERLESS



```
[david-dmn: ~ ⚡ clear  
[david-dmn: ~ ⚡ node -v  
v6.10.3  
david-dmn: ~ ⚡ npm install -g serverless
```

# CREATE SERVERLESS PROJECT

```
[david-dmn: serverless-workshop] ⚡ serverless create --template aws-nodejs --path serverless-workshop
Serverless: Generating boilerplate...
Serverless: Generating boilerplate in "/Users/Admin1/serverless-workshop/serverless-workshop"

┌───────────┐
│             │
│   ┌───┐     │
│   │   │     │
│   └───┘     │
│               │
│       The Serverless Application Framework
│               │
│           serverless.com, v1.26.0
│               │
│   ┌───┐     │
│   │   │     │
│   └───┘     │
│             │
└───────────┘

Serverless: Successfully generated boilerplate for template: "aws-nodejs"
[david-dmn: serverless-workshop] ⚡ █

[david-dmn: serverless-workshop] ⚡ cd serverless-workshop/
[david-dmn: serverless-workshop] ⚡ ls
handler.js      serverless.yml
[david-dmn: serverless-workshop] ⚡ █
```



# THE END

REPO: [HTTPS://GITHUB.COM/SUPERCYCLE91/SERVERLESS-  
WORKSHOP](https://github.com/supercycle91/serverless-workshop)

TWITTER: [@NOVICKI](#) DAVID

WE ARE HIRING!

## David Novicki