

# Up and Running with the Serverless Framework on AWS

---



**Fernando Medina Corey**

DATA ENGINEER

@fmc\_sea [www.fernandomc.com](http://www.fernandomc.com)



# Outline



## Installation

- Serverless installation
- OS differences
- AWS setup

## Serverless Application Model

- Functions, events, resources, and services

## Serverless Development

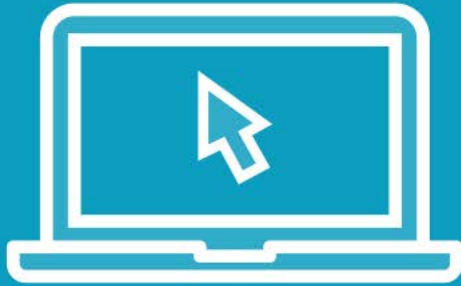
- Configuration, deployment, and testing

## Project

- Cron scheduled email reminder



# Demo



## Serverless Installation

- Prerequisites
  - AWS account setup
- Walkthroughs
  - Windows, OS X and Ubuntu
    - Node.js and npm
    - Serverless Framework
    - AWS CLI



# Serverless Application Model

---



# Serverless Application Components



Functions



Events



Resources



Services

# Serverless Functions



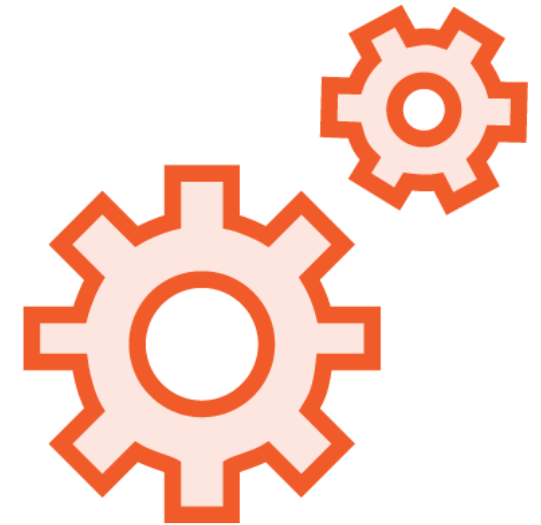
Open Whisk



Google Cloud  
Functions



Azure Functions



AWS Lambda

# Serverless Events (AWS)

API Gateway

Schedule

S3

CloudWatch

DynamoDB

Alexa Skills



# Serverless Resources (AWS)



DynamoDB



Simple Storage  
Service (S3)



CloudFront  
Distribution



Cognito User  
Pool



# Serverless Services

“A service is like a project. It's where you define your AWS Lambda Functions, the events that trigger them and any AWS infrastructure resources they require, all in a file called `serverless.yml`.”

- *serverless.com*



# Service Characteristics

Logical Organization

Shared Configuration

Specific REST APIs

Shared Resources



# What's in a Service?



serverless.yml



Functions



Dependencies

```
woofGarden/  
  serverless.yml  
  handler.js
```

---

```
woofGardenPets/  
  serverless.yml  
  handler.js  
  node_modules/  
  package.json  
woofGardenReminders/  
  serverless.yml  
  handler.js  
  node_modules/  
  package.json
```

## Dependencies & Structure

- ◀ Setup for a simple single-service app
- ◀ Medium-sized app with multiple services
- ◀ Service & DB Configuration
- ◀ CRUD functions in handler.js
- ◀ Dependencies in node\_modules
- ◀ package information details
- ◀ Service Configuration
- ◀ Reminder functions
- ◀ Service-specific dependencies



# serverless.yml

```
service:
  name: myService

provider:
  name: aws
  runtime: nodejs6.10

package:
  include:
    - src/**
    - handler.js
  exclude:
    - .git/**
    - .travis.yml
```

```
functions:
  usersCreate:
    handler: users.create
    description: My function
    memorySize: 512
    runtime: nodejs6.10
    timeout: 10
    role: arn:aws:iam::XXXXXX:role/role
    events:
      - schedule:
          rate: rate(10 minutes)
```

```
resources:
  Resources:
    usersTable:
      Type: AWS::DynamoDB::Table
      Properties:
        TableName: usersTable
        AttributeDefinitions:
          - AttributeName: email
            AttributeType: S
        KeySchema:
          - AttributeName: email
            KeyType: HASH
        ProvisionedThroughput:
          ReadCapacityUnits: 1
          WriteCapacityUnits: 1
```



# Functions

```
'use strict';

module.exports.hello = (event, context, callback) => {
  const response = {
    statusCode: 200,
    body: JSON.stringify({
      message: 'Your function executed successfully!',
      input: event,
    }),
  };

  callback(null, response);
};
```



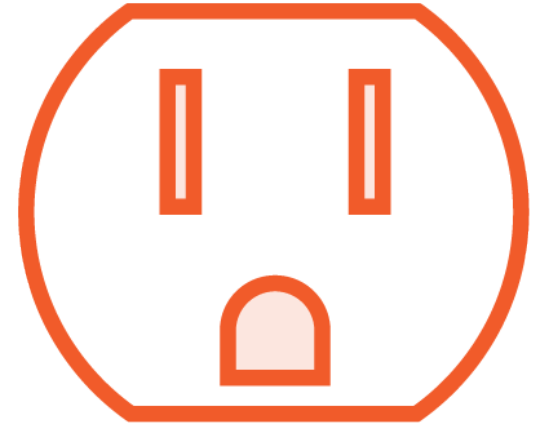
# Dependencies



Custom Code



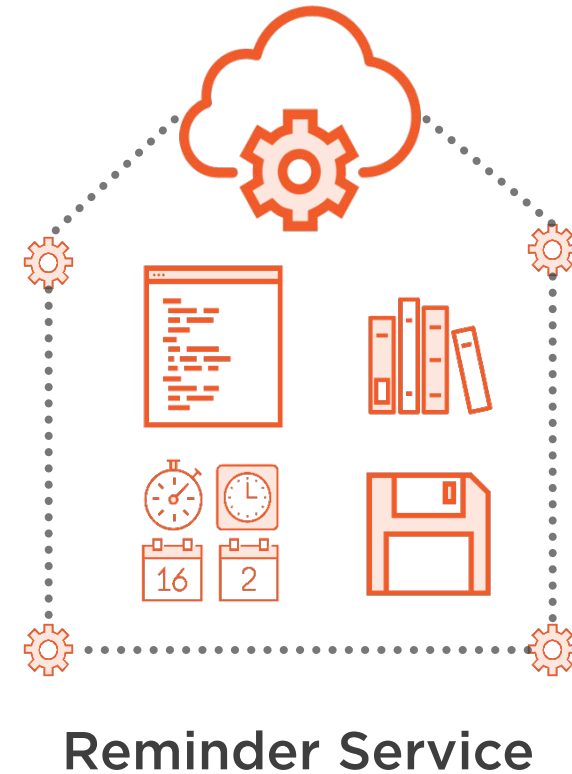
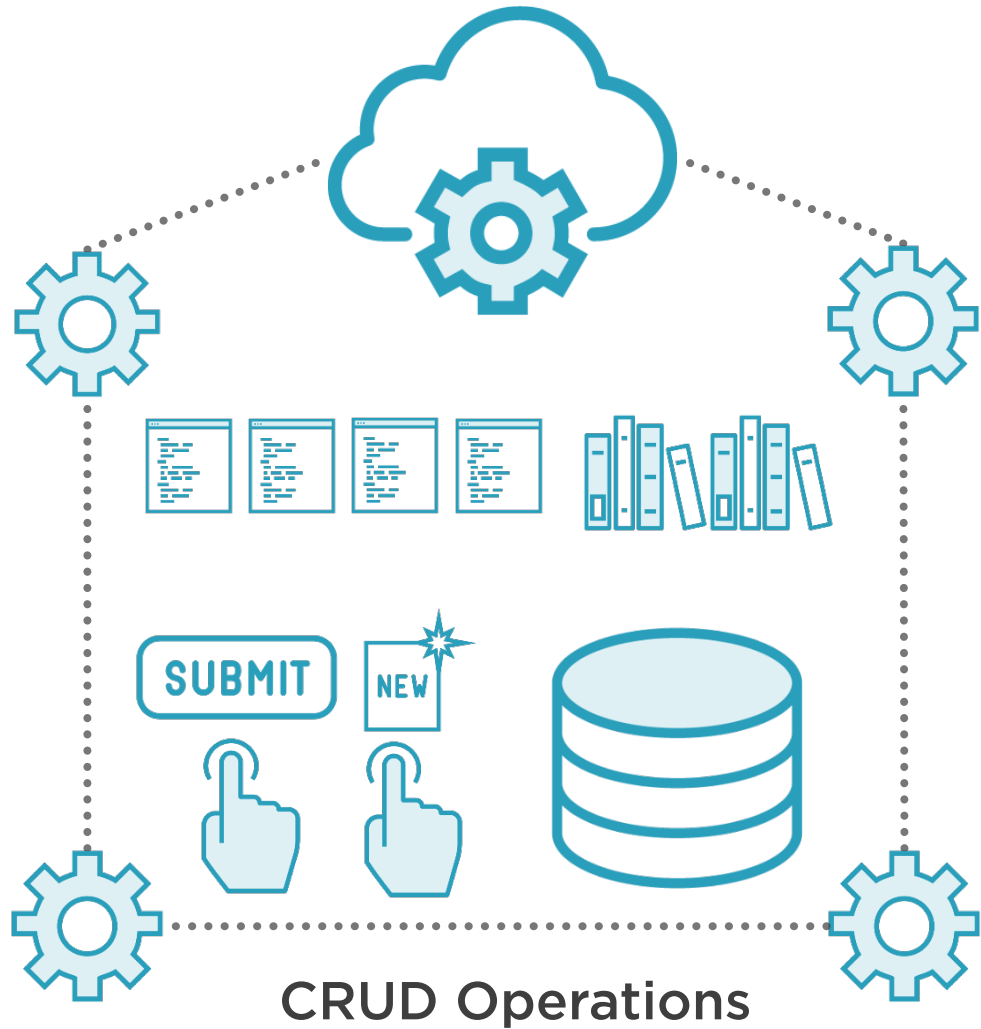
npm Packages



Serverless Plugins



# Serverless Services Visualized



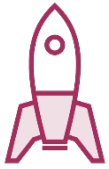


# Serverless Development

---



# Development Steps



**Setup and dependencies**



**Configuration**



**Write function code**



**Deploy and test**



# Setup and Dependencies

```
# Create a Serverless Framework service from a template  
serverless create --template aws-nodejs --path servicename
```

```
# Install any Node.js dependencies  
npm init && npm install your-dependencies
```



# Configuration - serverless.yml

```
# serverless.yml
service: woofGardenService

provider:
  name: aws
  runtime: nodejs6.10

functions:
  Emailer:
    handler: handler.woofGardenEmailer
    description: Emails reminders
  events:
    - schedule:
        rate: rate(3 days)
```

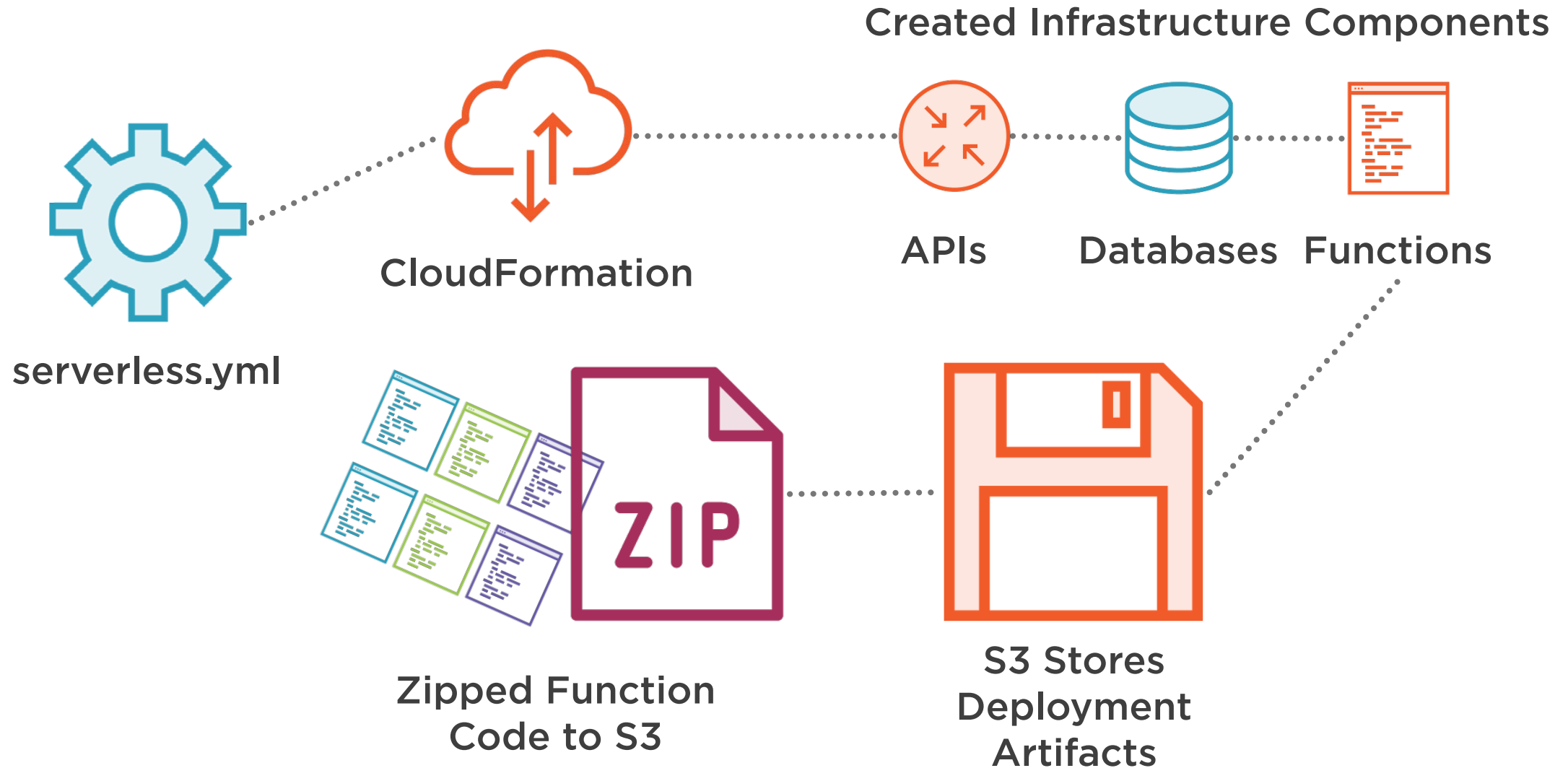




## Deployment Options

- serverless deploy
- sls deploy
- sls deploy function --function Emailer
- sls deploy --package package-path
- serverless remove

# Serverless Deployment



# Testing



Look ahead - Write testable function code



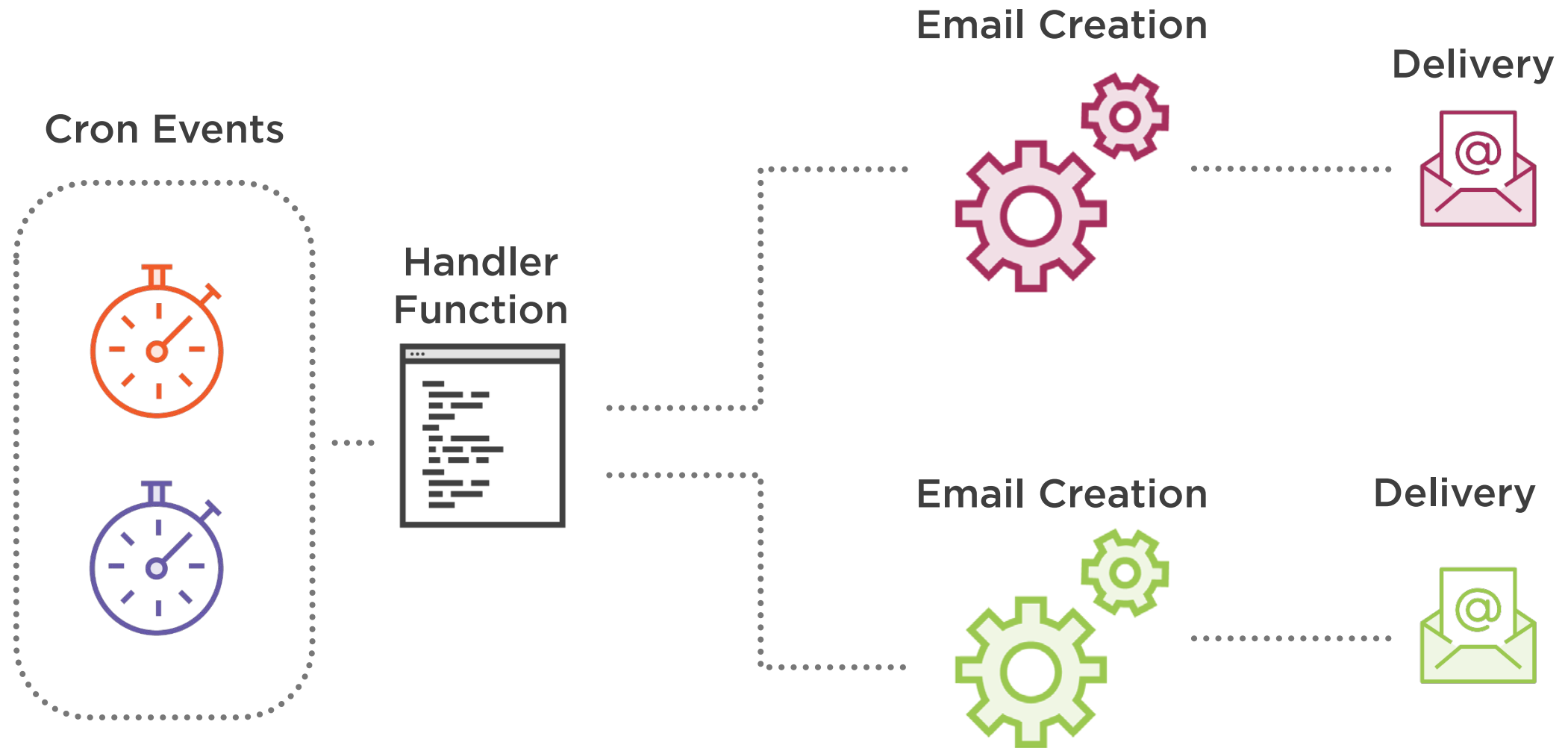
Separate business logic and IaaS-specific code



Unit tests for business logic and integration tests with other services

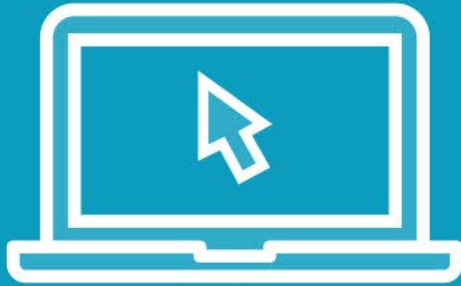


# Demo - Email Reminders





# Demo



## Creating our first Serverless Service

- Create service
- `serverless.yml`
- `handler.js`
- Setup SES
- Dependencies
- Deployment



# Review



## Serverless Concepts

- Functions
- Events
- Resources
- Services

## Serverless Skills

- Development
- Configuration
- Deployment
- Testing

## Coming Up

- Serverless APIs and applications

