



THE **CURVE**

 @THECURVECONSULTING



THECURVECONSULTING.IO



THE CURVE

Running Ruby and Rails, purely serverless on AWS

Paul & James Ridgway

The Speakers

Paul Ridgway

- CEO of The Curve
 - Work across the entire tech stack
 - Management & Leadership
 - Strategy
- Formerly CTO at The Floow

James Ridgway

- CTO of The Curve
 - Cross-language expert
 - Dev-ops mentality
 - Team leader
- Formerly Head of Platform and Head of Data Science at The Floow



About The Curve

What we do

- Bespoke software development
- Leadership and Strategy Consulting

Who we work with

- Playwerks
- Kollider Projects / Skkope.tv /
Kurious.art
- The University of Sheffield
- Useful Insurance
- Supercity Ltd
- Rekkommend



Overview

- What is Serverless?
 - Why?
 - Challenges?
- AWS Lambda
- Serverless Framework
- AWS CloudFormation
- AWS API Gateway
- Demos
- Basic Function
- HTTP Requests
- Rubygems
- Sinatra
- Rails
- Custom Domain
- Links



Demo Dependencies

- AWS Account
- AWS CLI / Profile Configuration
- npm install -g serverless
- npm install --save-dev serverless-ruby-package
- gem install jets
- npm install -g yarn
- ruby-2.5.3



What is Serverless?

- Serverless is a **concept**.
- The (cloud) **provider runs the hardware** and all intermediate software, and **manages the operational aspects**.
- Actual **resource consumption based pricing**, rather than up-front capacity purchase.
- Serverless products often **allow developers to run code**, along with providing other services like databases.



Why Serverless?

- **Cost**

Depending on the use case it can be cheaper.

- **Scaling**

Often managed by the provider.

- **Productivity**

Just worry about the code and logic required.

- **Reduced Overheads**

Reduced operational overhead / outsourced operations.



Serverless: Challenges

- **Performance**

“cold starts”

- **Resource limits**

Certain workloads are not suited to serverless

- **Debugging**

Slow logs, real testing requires deployments, no SSH

- **Security and Privacy**

A black-box system compared to classic compute, details and access provisions can be opaque

- **Standards and Vendor Lock-In**

A risk but a number of work arounds, for example docker



AWS Lambda

“AWS Lambda is an **event-driven, serverless computing platform** provided by Amazon as a part of the Amazon Web Services. It is a computing service that **runs code in response to events** and **automatically manages the computing resources** required by that code.”



AWS Lambda

- Supports Node.js, Python, Java, Go, Ruby, .NET Core and many others through an open source runtime model.
- Lamdas can be triggered by HTTP Requests, Schedules, other AWS Events or direct invocation.
- Billed per 100ms of compute time.
- Tiered by memory allocation (and CPU).
- Limited to 300 seconds of execution time.



AWS Lambda

Memory (MB)	Free tier seconds per month	Price per 100ms (\$)
128	3,200,000	0.000000208
192	2,133,333	0.000000313
256	1,600,000	0.000000417
320	1,280,000	0.000000521
384	1,066,667	0.000000625
448	914,286	0.000000729
512	800,000	0.000000834
576	711,111	0.000000938
640	640,000	0.000001042
704	581,818	0.000001146
768	533,333	0.000001250
832	492,308	0.000001354
896	457,143	0.000001459
960	426,667	0.000001563
1024	400,000	0.000001667
1088	376,471	0.000001771
1152	355,556	0.000001875
1216	336,842	0.000001980



Lambda Management Console

https://eu-west-2.console.aws.amazon.com/lambda/home?region=eu-west-2#/functions/gems-function-dev-hello?tab=graph

aws

Services

Resource Groups

Paul @ paulridgway

London

Support

Lambda > Functions > gems-function-dev-hello

ARN - arn:aws:lambda:eu-west-2:181984840591:function:gems-function-dev-hello

gems-function-dev-hello

Throttle

Qualifiers

Actions

Select a test event

Test

Save

This function belongs to the AWS CloudFormation stack **gems-function-dev**. [Manage this stack](#) on the CloudFormation console.

Configuration

Monitoring

▼ Designer

Add triggers

Choose a trigger from the list below to add it to your function.

API Gateway

AWS IoT

Application Load Balancer

CloudWatch Events

CloudWatch Logs

CodeCommit

Go back to application gems-function-dev

gems-function-dev-hello

Layers (0)

API Gateway

Add triggers from the list on the left

Amazon CloudWatch Logs

Resources that the function's role has access to appear here

Function code

Info

Feedback

English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Lambda Management Console

https://eu-west-1.console.aws.amazon.com/lambda/home?region=eu-west-1#/functions/billing-alerts_notify?tab=graph

aws

Services

Resource Groups

bell

paul @ paulridgway

Ireland

Support

billing-alerts_notify

Throttle

Qualifiers

Actions

test

Test

Save

Environment

billing-alerts_notify

vendor

Gemfile

Gemfile.lock

lambda.rb

log_helper.rb

notify.rb

template.html

lambda.rb

1require 'securerandom'

2require 'logger'

3require_relative 'log_helper'

4require_relative 'notify'

5

6\$lambda = !!ENV['AWS_LAMBDA_FUNCTION_NAME']

7

8L = LogHelper::logger("lambda-#{SecureRandom.uuid[-6...-1]}")

9

10NOTIFY = Notify.new

11

12def handler(event:, context:)

13 L.info("Request received, event: #{JSON.dump(event)}, context: #{JSON.dump(context)}")

14

15 # The event payload will look something like this

16 # {

17 # "version": "0",

18 # "id": "47df9fb6-e2dc-b381-7201-2ff06e9579af",

19 # "detail-type": "Scheduled Event",

20 # "source": "aws.events",

21 # "account": "181984840591",

22 # "time": "2019-02-10T13:55:32Z",

23 # "region": "eu-west-1",

24 # "resources": ["arn:aws:events:eu-west-1:181984840591:rule/billing-alert"],

25 # "detail": {}

26 # }

27

28 NOTIFY.run

29end

30

31L.info "Checking environment"

32if \$lambda

33 L.info "Running in lambda"

1:1 Ruby Spaces: 2

Environment variables

Feedback

English (US)

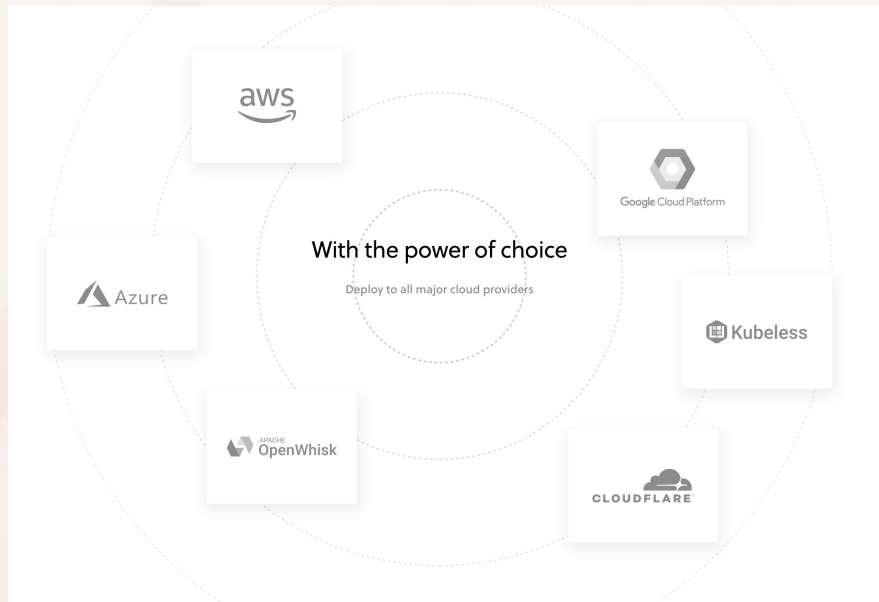
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Serverless Framework

“The Serverless Framework is a free and open-source web framework written using Node.js. Serverless is the first framework that was originally developed for building applications exclusively on AWS Lambda, a serverless computing platform provided by Amazon as a part of the Amazon Web Services.”

Supports AWS, Azure, GCP and many others.

Lots of plugins (eg for Ruby Gems).



Serverless Framework

```
# Create a new Serverless Service/Project
$ serverless create --template aws-nodejs --path my-service
```

```
# Change into the newly created directory
$ cd my-service
```

```
# Deploy "everything"
$ serverless deploy -v
```

```
# Deploy the function code only
$ serverless deploy function -f hello
```

```
# Run the function
$ serverless invoke -f hello -l
```

```
# Tail the logs
$ serverless logs -f hello -t
```



Serverless Framework

```
1  service: basic-function
2
3  provider:
4    name: aws
5    runtime: ruby2.5
6    region: eu-west-2
7
8  functions:
9    hello:
10     handler: handler.hello
11     events:
12       - http:
13         path: hello
14         method: post
15       - http:
16         path: hello
17         method: get
```

```
1  require 'json'
2
3  def hello(event:, context:)
4    { statusCode: 200, body: JSON.pretty_generate(event) }
5  end
```



AWS CloudFormation

“AWS CloudFormation provides a common language for you to **describe and provision all the infrastructure resources** in your cloud environment. CloudFormation allows you to use a **simple text file to model and provision**, in an automated and secure manner, all the resources needed for your applications across all regions and accounts. This file serves as the **single source of truth for your cloud environment**”



AWS CloudFormation

- Describe a deployment, service or environment with a single YAML or JSON file.
- Can provision and reference any type of resource
- Resources of a deployment are tracked, they can all be deleted in one go. Detects configuration drift
- AWS-only. No additional charge
- Used by the Serverless Framework



CloudFormation - Stack rails-f...

https://eu-west-2.console.aws.amazon.com/cloudformation/home?region=eu-west-2#/stacks/resources?filter=active&stackId=arn...

aws

Services

Resource Groups

bell

paul @ paulridgway

London

Support

CloudFormation > Stacks: rails-function-dev

Stacks (4)

Create stack

Active

View nested

1

rails-function-dev

Sat, 11 May 2019 15:49:16 GMT

UPDATE_COMPLETE

sinatra-function-dev

Sat, 11 May 2019 13:26:51 GMT

UPDATE_COMPLETE

gems-function-dev

Sat, 11 May 2019 10:42:43 GMT

UPDATE_COMPLETE

basic-function-dev

Sat, 11 May 2019 08:04:32 GMT

UPDATE_COMPLETE

rails-function-dev

Delete

Update

Stack actions

Stack info

Events

Resources

Outputs

Parameters

Template

Resources (13)

Search resources

Logical ID	Physical ID	Type	Status	Status reason
ServerlessDeploymentBucket	rails-function-dev-serverlessdeploymentbucket-dftyqtl6zmmj	AWS::S3::Bucket	CREATE_COMPLETE	-
IamRoleLambdaExecution	rails-function-dev-eu-west-2-lambdaRole	AWS::IAM::Role	CREATE_COMPLETE	-
HelloLogGroup	/aws/lambda/rails-function-dev-hello	AWS::Logs::LogGroup	CREATE_COMPLETE	-
HelloLambdaVersion5MuWND165TqPx0Vd8rG3Vwop2oe4sXPUTmjC3c3nNy k	arn:aws:lambda:eu-west-2:181984840591:function:rails-function-dev-hello:27	AWS::Lambda::Version	CREATE_COMPLETE	-
HelloLambdaPermissionApiGateway	rails-function-dev-HelloLambdaPermissionApiGateway-SFDON52YQLVB	AWS::Lambda::Permission	CREATE_COMPLETE	-
HelloLambdaFunction	rails-function-dev-hello	AWS::Lambda::Function	CREATE_COMPLETE	-

https://eu-west-2.console.aws.amazon.com/cloudformation/home?region=eu-west-2#

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

AWS API Gateway

“Amazon API Gateway is a **fully managed service** that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create **REST** and **WebSocket APIs** that act as a “**front door**” for applications to access data, business logic, or functionality from your backend services, such as workloads running on Amazon Elastic Compute Cloud (**Amazon EC2**), code running on **AWS Lambda**, any web application, or real-time communication applications.”



AWS API Gateway

- Serverless mapping from HTTP(s) requests to Lambda
- Per request pricing, no load-balancer cost
- Custom domain support via CloudFront



API Gateway

https://eu-west-2.console.aws.amazon.com/apigateway/home?region=eu-west-2#/apis/prv1ch1on6/resources/1g3n3j/methods/GET

ServicesResource Groups

Paul @ paulridgwayLondonSupport

Amazon API GatewayAPIs > dev-sinatra-function (prv1ch1on6) > Resources > /{path+} (1g3n3j) > GETShow all hints?

APIs

dev-basic-functiondev-gems-functiondev-rails-functiondev-sinatra-functionResourcesStagesAuthorizersGateway ResponsesModelsResource PolicyDocumentationDashboardSettingsUsage PlansAPI KeysCustom Domain NamesClient CertificatesVPC Links

Resources

/{path+}DELETEGETPOSTPUT

Actions

TEST

/{path+} - GET - Method Execution

Client

Method Request

Auth: NONEARN: arn:aws:execute-api:eu-west-2:181984840591:prv1ch1on6/*/GET/*

Integration Request

Type: LAMBDA_PROXY

Method Response

Select an integration response.

Integration Response

Proxy integrations cannot be configured to transform responses.

Lambda sinatra-function-dev-hello

FeedbackEnglish (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy PolicyTerms of Use

Demos / Walkthroughs



Basic Ruby Function

Basic Ruby Function: Configuration

```
service: aws-ruby
```

```
provider:
```

```
  name: aws
```

```
  runtime: ruby2.5
```

```
  region: eu-west-2
```

```
functions:
```

```
  hello:
```

```
    handler: handler.hello
```



Basic Ruby Function: Deploy

```
paul@pdr [10:13:25] [~/Documents/shrug/basic-function]
-> % sls deploy
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Creating Stack...
Serverless: Checking Stack create progress...
.....
Serverless: Stack create finished...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service aws-ruby.zip file to S3 (266 B)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.....
```

Basic Ruby Function: Deploy

Serverless: Stack update finished...

Service Information

service: aws-ruby

stage: dev

region: eu-west-2

stack: aws-ruby-dev

resources: 5

api keys:

None

endpoints:

None

functions:

hello: aws-ruby-dev-hello

layers:

None

Lambda Management Console

https://eu-west-2.console.aws.amazon.com/lambda/home?region=eu-west-2#/functions/aws-ruby-dev-hello?tab=graph

aws

Services

Resource Groups

bell

paul @ paulridgway

London

Support

Lambda > Functions > aws-ruby-dev-hello

ARN - arn:aws:lambda:eu-west-2:181984840591:function:aws-ruby-dev-hello

aws-ruby-dev-hello

Throttle

Qualifiers

Actions

test

Test

Save

Execution result: succeeded (logs)

Details

The section below shows the result returned by your function execution.

```
{
  "statusCode": 200,
  "body": "\"Go Serverless v1.0! Your function executed successfully!\""
}
```

Summary

Code SHA-256	Request ID
9wdrUTi8cgWzkQAMNZVmkdpdPplDsj2o2R1gKJs7UfM=	51990feb-7317-49f5-aef2-05f683e4e8ae
Duration	Billed duration
1.83 ms	100 ms
Resources configured	Max memory used
1024 MB	46 MB

Log output

The section below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. [Click here](#) to view the CloudWatch log group.

```
START RequestId: 51990feb-7317-49f5-aef2-05f683e4e8ae Version: $LATEST
END RequestId: 51990feb-7317-49f5-aef2-05f683e4e8ae
REPORT RequestId: 51990feb-7317-49f5-aef2-05f683e4e8ae  Duration: 1.83 ms    Billed Duration: 100 ms    Memory Size: 1024 MB
Max Memory Used: 46 MB
```

Feedback

English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Accept HTTP(s) Requests

```
service: aws-ruby
```

```
provider:
```

```
  name: aws
```

```
  runtime: ruby2.5
```

```
  region: eu-west-2
```

```
functions:
```

```
  hello:
```

```
    handler: handler.hello
```

```
    events:
```

```
      - http:
```

```
        path: test
```

```
        method: get
```



Accept HTTP(s) Requests

```
paul@pdr [10:20:10] [~/Documents/shrug/basic-function]
-> % sls deploy
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service aws-ruby.zip file to S3 (9.35 MB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.....
```

Accept HTTP(s) Requests

```
Serverless: Stack update finished...
```

```
Service Information
```

```
service: aws-ruby
```

```
stage: dev
```

```
region: eu-west-2
```

```
stack: aws-ruby-dev
```

```
resources: 10
```

```
api keys:
```

```
  None
```

```
endpoints:
```

```
  GET - https://somerandomid.execute-api.eu-west-2.amazonaws.com/dev/test
```

```
functions:
```

```
  hello: aws-ruby-dev-hello
```

```
layers:
```

```
  None
```


Accept HTTP(s) Requests

```
paul@pdr [10:20:46] [~/Documents/shrug/basic-function]
-> % curl 'https://somerandomid.execute-api.eu-west-2.amazonaws.com/dev/test'
"Go Serverless v1.0! Your function executed successfully!"
```

Supporting Gems

```
paul@pdr [10:25:51] [~/Documents/shrug/basic-function]
-> % bundle init
Writing new Gemfile to /Users/paul/Documents/shrug/basic-function/Gemfile
```

Supporting Gems

```
# frozen_string_literal: true
```

```
source "https://rubygems.org"
```

```
git_source(:github) {|repo_name| "https://github.com/#{repo_name}" }
```

```
gem "coderay"
```



Supporting Gems

```
paul@pdr [10:26:15] [~/Documents/shrug/basic-function]
-> % bundle install
Fetching gem metadata from https://rubygems.org/.
Resolving dependencies...
Using bundler 1.17.2
Fetching coderay 1.1.2
Installing coderay 1.1.2
Bundle complete! 1 Gemfile dependency, 2 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
```

Supporting Gems

```
require 'coderay'  
require 'json'
```

```
def hello(event:, context:)  
  { statusCode: 200, body: CodeRay.scan(File.read(__FILE__), :ruby).div }  
end
```



Supporting Gems

```
paul@pdr [10:28:21] [~/Documents/shrug/basic-function]
-> % sls invoke local -f hello
{"statusCode":200,"body":"<div class=\"CodeRay\">\n  <div class=\"code\"><pre>require <span
style=\"background-color:hsla(0,100%,50%,0.05)\"><span style=\"color:#710\">'</span><span
style=\"color:#D20\">coderay</span><span style=\"color:#710\">'</span></span></span>\nrequire
<span style=\"background-color:hsla(0,100%,50%,0.05)\"><span
style=\"color:#710\">'</span><span style=\"color:#D20\">json</span><span
style=\"color:#710\">'</span></span></span>\n\n<span
style=\"color:#080;font-weight:bold\">def</span> <span
style=\"color:#06B;font-weight:bold\">hello</span>(<span
style=\"color:#606\">event</span>:, <span style=\"color:#606\">context</span>:)\n  { <span
style=\"color:#606\">statusCode</span>: <span style=\"color:#00D\">200</span>, <span
style=\"color:#606\">body</span>: <span
style=\"color:#036;font-weight:bold\">CodeRay</span>.scan(<span
style=\"color:#036;font-weight:bold\">File</span>.read(<span
style=\"color:#069\">__FILE__</span>), <span style=\"color:#A60\">:ruby</span>).div
}\n<span style=\"color:#080;font-weight:bold\">end</span>\n</pre></div>\n</div>\n"}
~
```

Supporting Gems

```
paul@pdr [10:29:06] [~/Documents/shrug/basic-function]
-> % sls logs -f hello -t
START RequestId: 063d532b-84fd-4eb3-9d15-e648b10f0306 Version: $LATEST
Init error when loading handler handler.hello
{
  "errorMessage": "cannot load such file -- coderay",
  "errorType": "Init<LoadError>",
  "stackTrace": [
    "/var/lang/lib/ruby/site_ruby/2.5.0/rubygems/core_ext/kernel_require.rb:54:in
`require'",
    "/var/lang/lib/ruby/site_ruby/2.5.0/rubygems/core_ext/kernel_require.rb:54:in
`require'",
    "/var/task/handler.rb:1:in `
```

Supporting Gems

```
paul@pdr [10:14:31] [~/Documents/shrug/basic-function]
-> % npm i --save-dev serverless-ruby-package
npm WARN saveError ENOENT: no such file or directory, open
'/Users/paul/Documents/shrug/basic-function/package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open
'/Users/paul/Documents/shrug/basic-function/package.json'
npm WARN basic-function No description
npm WARN basic-function No repository field.
npm WARN basic-function No README data
npm WARN basic-function No license field.

+ serverless-ruby-package@1.1.1
added 1 package in 2.888s
```


Supporting Gems

...

functions:

 hello:

 handler: handler.hello

 events:

 - http:

 path: test

 method: get

package:

 include:

 - handler.rb

plugins:

 - serverless-ruby-package



Supporting Gems

```
-> % bundle install --standalone --path vendor/bundle
Fetching gem metadata from https://rubygems.org/.
Using bundler 1.17.2
Fetching coderay 1.1.2
Installing coderay 1.1.2
Bundle complete! 1 Gemfile dependency, 2 gems now installed.
Bundled gems are installed into `./vendor/bundle`
```

Supporting Gems

```
load "vendor/bundle/bundler/setup.rb"
```

```
require 'coderay'
```

```
require 'json'
```

```
def hello(event:, context:)
```

```
  { statusCode: 200, body: CodeRay.scan(File.read(__FILE__), :ruby).div }  
end
```



Supporting Gems

```
paul@pdr [10:34:04] [~/Documents/shrug/basic-function]
-> % sls deploy
ruby-package: Packaging gems: coderay-1.1.2
Serverless: Packaging service...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service aws-ruby.zip file to S3 (645 B)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.....
```

Supporting Gems

```
Serverless: Stack update finished...
Service Information
service: aws-ruby
stage: dev
region: eu-west-2
stack: aws-ruby-dev
resources: 10
api keys:
  None
endpoints:
  GET - https://2w718clk16.execute-api.eu-west-2.amazonaws.com/dev/test
functions:
  hello: aws-ruby-dev-hello
layers:
  None
```

Supporting Gems

```
load "vendor/bundle/bundler/setup.rb"
```

```
require 'coderay'
```

```
require 'json'
```

```
def hello(event:, context:)
```

```
{
```

```
  statusCode: 200,
```

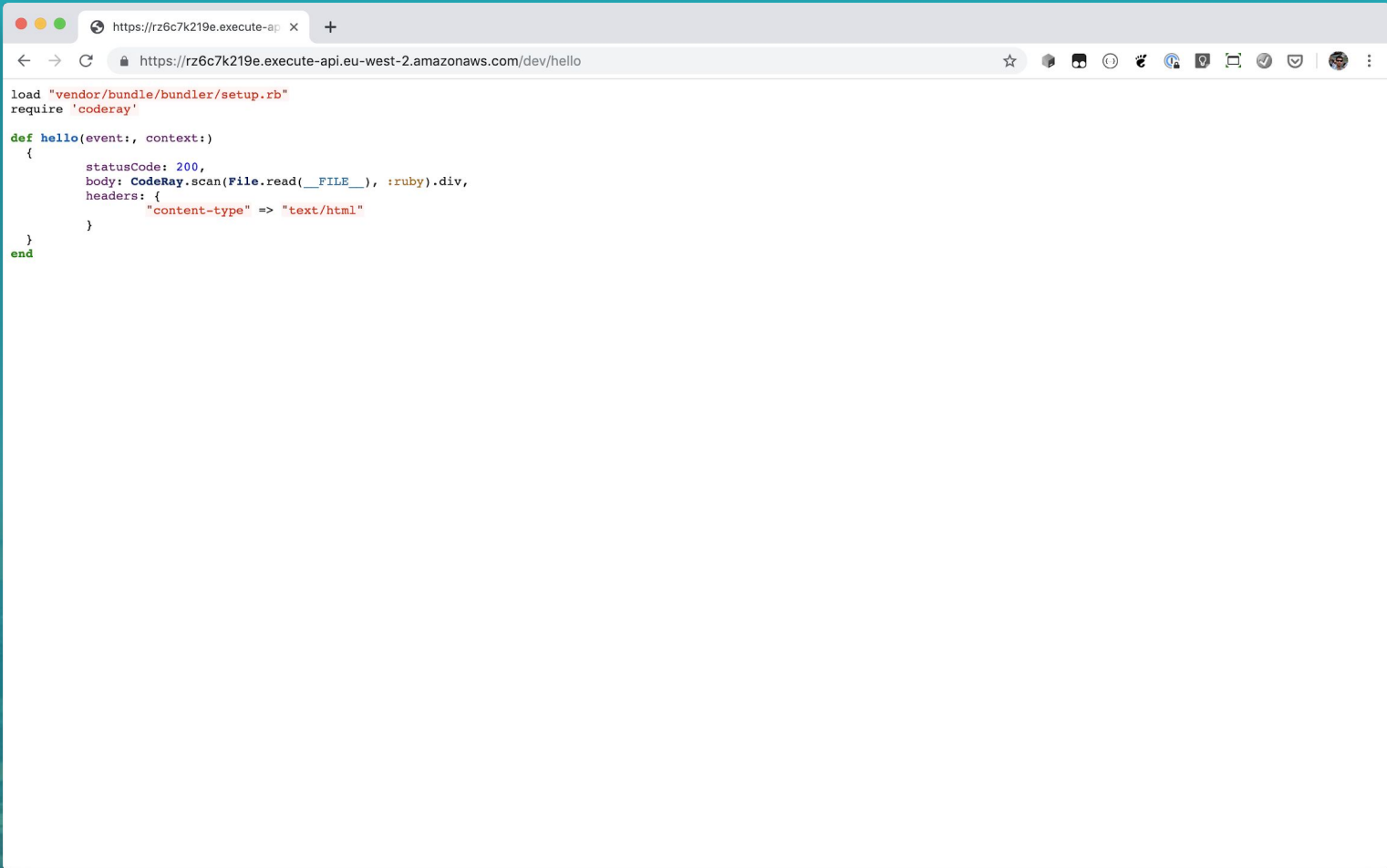
```
  body: CodeRay.scan(File.read(__FILE__), :ruby).div ,
```

```
  headers: {"content-type" => "text/html"}
```

```
}
```

```
end
```





Sinatra App / API: Gemfile

```
source 'https://rubygems.org'  
gem 'sinatra'  
gem 'json'  
gem 'rack'  
gem 'rack-contrib'  
gem 'rake'
```



Sinatra App / API: serverless.yml

```
functions:
  hello:
    handler: lambda.handler
    events:
      - http:
          path: /{path+}
          method: get
      - http:
          path: /{path+}
          method: post
```

```
package:
  include:
    - lambda.rb
    - app/**
```

```
plugins:
  - serverless-ruby-package
```



Sinatra App / API: lambda.rb

```
load "vendor/bundle/bundler/setup.rb"
require 'json'
require 'rack'

$app ||= Rack::Builder.parse_file("#{File.dirname(__FILE__)}/app/config.ru").first

def handler(event:, context:)
  path = (event['path'] || "").gsub(/^\//app/, '')
  env = {
    "REQUEST_METHOD" => event['httpMethod'],
    "SCRIPT_NAME" => "",
    "PATH_INFO" => path,
    "QUERY_STRING" => event['queryStringParameters'] || "",
    "SERVER_NAME" => "localhost",
    "SERVER_PORT" => 443,
    "rack.version" => Rack::VERSION,
    "rack.url_scheme" => "https",
    "rack.input" => StringIO.new(event['body'] || ""),
    "rack.errors" => $stderr,
  }
  unless event['headers'].nil?
    event['headers'].each{ |key, value| env["HTTP_#{key}"] = value }
  end
end
```



Sinatra App / API: lambda.rb

...

```
begin
  status, headers, body = $app.call(env)

  body_content = ""
  body.each do |item|
    body_content += item.to_s
  end

  response = {
    "statusCode" => status,
    "headers" => headers,
    "body" => body_content
  }
rescue Exception => msg
  response = {
    "statusCode" => 500,
    "body" => msg
  }
end
response
end
```



Sinatra App / API: app/config.ru

```
require 'rack'
require 'rack/contrib'
require_relative '../server'

set :root, File.dirname(__FILE__)
set :views, Proc.new { File.join(root, "views") }

run Sinatra::Application
```



Sinatra App / API: app/server.rb

```
require 'sinatra'

before do
  if request.body.size > 0
    request.body.rewind
    @params = Sinatra::IndifferentHash.new
    @params.merge!(JSON.parse(request.body.read))
  end
end

get '/' do
  erb :index
end

get '/hello-world' do
  content_type :json
  { :Output => 'Hello World!' }.to_json
end

post '/hello-world' do
  content_type :json
  { :Output => 'Hello World!' }.to_json
end
```



Sinatra App / API: app/views/index.erb

```
<html>
  <head>
    <title>Hello!</title>
  </head>
  <body>
    Hello!
  </body>
</html>
```



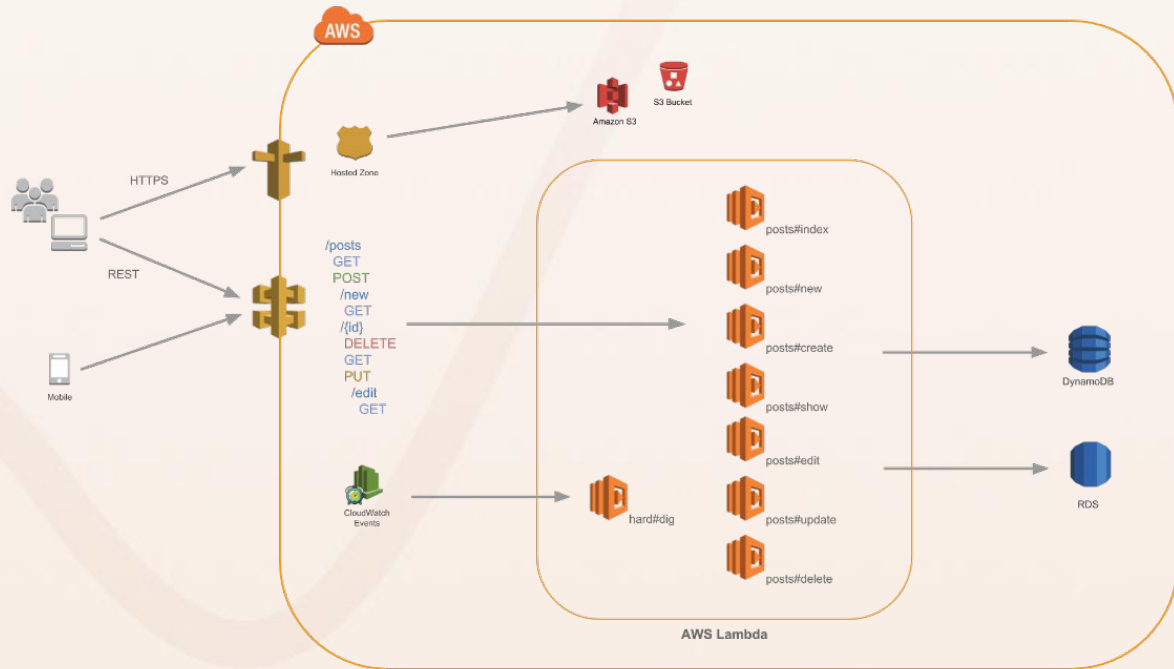
Sinatra App / API: app/views/index.erb

```
paul@aero15 [10:50:13] [~/Documents/Code/serverless-ruby-sinatra] [master *]  
-> % curl https://prvlch1on6.execute-api.eu-west-2.amazonaws.com/dev/app/hello-world  
{"Output":"Hello World!"}%
```

```
paul@aero15 [10:52:12] [~/Documents/Code/serverless-ruby-sinatra] [master *]  
-> % curl https://prvlch1on6.execute-api.eu-west-2.amazonaws.com/dev/app  
<html>  
  <head>  
    <title>Hello!</title>  
  </head>  
  <body>  
    Hello!  
  </body>  
</html>
```

Rails: Ruby on Jets

“Ruby on Jets allows you to create and deploy serverless services with ease, and to seamlessly glue AWS services together with the most beautiful dynamic language: Ruby. It includes everything you need to build an API and deploy it to AWS Lambda. Jets leverages the power of Ruby to make serverless joyful for everyone.”



Rails: Ruby on Jets

```
paul@pdr [11:28:23] [~/Documents/shrug]
-> % jets new --no-database jets-demo
Creating new project called jets-demo.
...

paul@pdr [11:29:09] [~/Documents/shrug]
-> % cd jets-demo

paul@pdr [11:30:39] [~/Documents/shrug/jets-demo] [master]
-> % bundle install

paul@pdr [11:30:42] [~/Documents/shrug/jets-demo] [master]
-> % jets server
=> bundle exec shotgun --port 8888 --host 127.0.0.1
Jets booting up in development mode!
== Shotgun/WEBrick on http://127.0.0.1:8888/
[2019-05-13 11:30:56] INFO WEBrick 1.4.2
[2019-05-13 11:30:56] INFO ruby 2.5.3 (2018-10-18) [x86_64-darwin18]
```

Rails: Ruby on Jets

```
paul@pdr [11:31:18] [~/Documents/shrug/jets-demo] [master]
-> % jets generate controller demo
subl .      create  app/controllers/demo_controller.rb
           invoke  erb
           create   app/views/demo
           invoke   helper
           create   app/helpers/demo_helper.rb
```

Rails: Ruby on Jets

```
class DemoController < ApplicationController

  def index
  end

end
```



Rails: Ruby on Jets

```
Jets.application.routes.draw do  
  get "demo", to: "demo#index"  
  root "demo#index"  
end
```



Rails: Ruby on Jets

```
paul@pdr [11:45:10] [~/Documents/shrug/jets-demo] [master *]
-> % jets deploy
Deploying to Lambda jets-demo-dev environment...
Building CloudFormation templates.
Deploying CloudFormation stack with jets app!
11:45:16AM CREATE_IN_PROGRESS AWS::CloudFormation::Stack jets-demo-dev User Initiated
11:45:19AM CREATE_IN_PROGRESS AWS::S3::Bucket S3Bucket
11:45:20AM CREATE_IN_PROGRESS AWS::S3::Bucket S3Bucket Resource creation Initiated
11:45:41AM CREATE_COMPLETE AWS::S3::Bucket S3Bucket
11:45:43AM CREATE_COMPLETE AWS::CloudFormation::Stack jets-demo-dev
Stack success status: CREATE_COMPLETE
Time took for stack deployment: 31s.
```

Rails: Ruby on Jets

```
paul@pdr [11:45:10] [~/Documents/shrug/jets-demo] [master *]
-> % jets deploy
Deploying to Lambda jets-demo-dev environment...
Building CloudFormation templates.
Deploying CloudFormation stack with jets app!
11:45:16AM CREATE_IN_PROGRESS AWS::CloudFormation::Stack jets-demo-dev User Initiated
11:45:19AM CREATE_IN_PROGRESS AWS::S3::Bucket S3Bucket
11:45:20AM CREATE_IN_PROGRESS AWS::S3::Bucket S3Bucket Resource creation Initiated
11:45:41AM CREATE_COMPLETE AWS::S3::Bucket S3Bucket
11:45:43AM CREATE_COMPLETE AWS::CloudFormation::Stack jets-demo-dev
Stack success status: CREATE_COMPLETE
Time took for stack deployment: 31s.
...
```

Rails: Ruby on Jets

```
...
11:48:54AM CREATE_IN_PROGRESS AWS::CloudFormation::Stack DemoController Resource creation
Initiated
11:48:54AM CREATE_IN_PROGRESS AWS::CloudFormation::Stack JetsPreheatJob Resource creation
Initiated
11:49:28AM CREATE_COMPLETE AWS::CloudFormation::Stack DemoController
11:49:31AM CREATE_IN_PROGRESS AWS::CloudFormation::Stack ApiDeployment20190513114816
11:49:32AM CREATE_IN_PROGRESS AWS::CloudFormation::Stack ApiDeployment20190513114816
Resource creation Initiated
11:49:43AM CREATE_COMPLETE AWS::CloudFormation::Stack ApiDeployment20190513114816
11:50:49AM CREATE_COMPLETE AWS::CloudFormation::Stack JetsPreheatJob
11:50:52AM UPDATE_COMPLETE_CLEANUP_IN_PROGRESS AWS::CloudFormation::Stack jets-demo-dev
11:50:53AM UPDATE_COMPLETE AWS::CloudFormation::Stack jets-demo-dev
Stack success status: UPDATE_COMPLETE
Time took for stack deployment: 2m 30s.
Prewarming application.
API Gateway Endpoint: https://j5s9igfbgk.execute-api.eu-west-1.amazonaws.com/dev/
```

CloudFormation - Stack jets-d

https://eu-west-1.console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/stackinfo?stackId=arn%3Aaws%3Acl...

awsServicesResource Groups

bellpaul @ paulridgwayIrelandSupport

CloudFormation

Stacks

Stack details

Change sets

Drifts

StackSets

Exports

Designer

Previous console

Feedback

CloudFormation > Stacks: jets-demo-dev-ApiDeployment20190513114816-8SZV23WXJQVO

Stacks (8)

Create stack

Active

View nested < 1 >

NESTED

jets-demo-dev-ApiDeploymen

t20190513114816-8SZV23W

XJQVO

Mon, 13 May 2019 10:49:32 G

MT

CREATE_COMPLETE

NESTED

jets-demo-dev-JetsPreheatJob

-GGPBX83RM2ZF

Mon, 13 May 2019 10:48:54 GM

T

CREATE_COMPLETE

NESTED

jets-demo-dev-DemoControlle

r-ZNPI8B25PKOT

Mon, 13 May 2019 10:48:54 GM

T

CREATE_COMPLETE

NESTED

jets-demo-dev-ApiGateway-1

KRX5EV9EX0QC

jets-demo-dev-ApiDeployment20190513114816-8SZV23WXJQVO

NESTED

DeleteUpdateStack actions

Stack infoEventsResourcesOutputsParametersTemplate

Overview

Stack ID

arn:aws:cloudformation:eu-west-1:181984840591:stack/jets-demo-dev-ApiDeployment20190513114816-8SZV23WXJQVO/ca3bb860-756c-11e9-a327-0a675d25b48e

Description

-

Status

CREATE_COMPLETE

Status reason

-

Root stack

arn:aws:cloudformation:eu-west-1:181984840591:stack/jets-demo-dev/31b0f830-756c-11e9-b123-062647984902

Parent stack

arn:aws:cloudformation:eu-west-1:181984840591:stack/jets-demo-dev/31b0f830-756c-11e9-b123-062647984902

Created time

Mon, 13 May 2019 10:49:32 GMT

Deleted time

-

Updated time

-

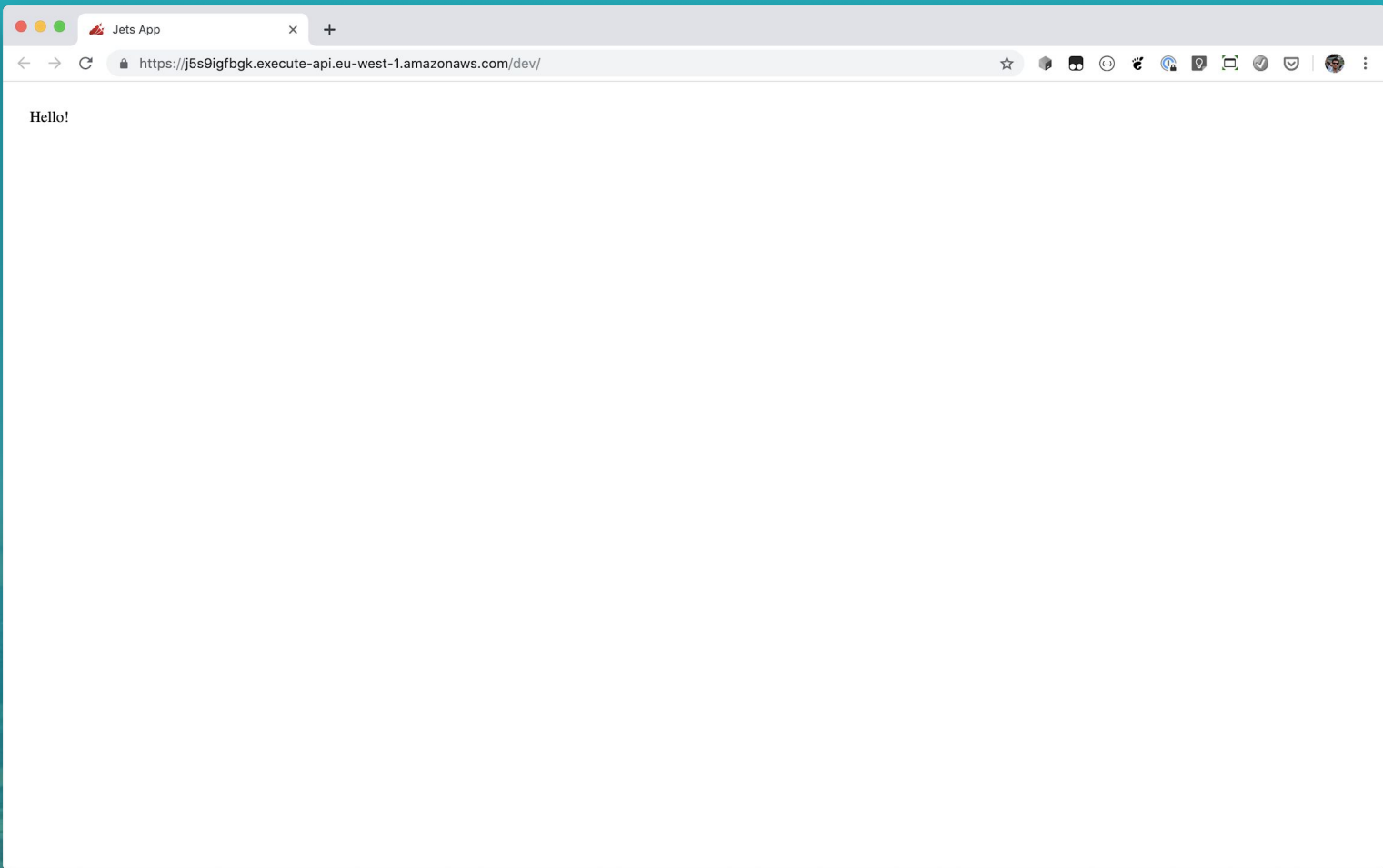
Last drift check time

Drift status

Feedback

English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Custom Domain

1. Create CloudFront distribution pointing to API Gateway.
2. Create an SSL Certificate.
3. Assign the SSL and CNAME to the CloudFront Distribution.
4. Add a DNS Alias to point to the Distribution.



Step 1: Select delivery method

Step 2: Create distribution

Create Distribution

Origin Settings

Origin Domain Name

sinatra-function-dev-serverlessdeployme

i

Origin Path

/dev/app

i

Origin ID

S3-sinatra-function-dev-serverlessdeplo

i

Restrict Bucket Access

☐ Yes

☒ No

i

Origin Custom Headers

Header Name	Value

i

+

Default Cache Behavior Settings

Path Pattern

Default (*)

i

Viewer Protocol Policy

☐ HTTP and HTTPS

☒ Redirect HTTP to HTTPS

☐ HTTPS Only

i

Allowed HTTP Methods

☐ GET, HEAD

☐ GET, HEAD, OPTIONS

☒ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

i

Field-level Encryption Config

i

Cached HTTP Methods

☒ GET, HEAD (Cached by default)

☐ OPTIONS

i

Cache Based on Selected Request Headers

None (Improves Caching)

i

Feedback

English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Dashboard

Hosted zones

Health checks

Traffic flow

Traffic policies

Policy records

Domains

Registered domains

Pending requests

Resolver

VPCs

Inbound endpoints

Outbound endpoints

Rules

Back to Hosted Zones

Create Record Set

Import Zone File

Delete Record Set

Test Record Set

Record Set Name

Any Type

Aliases Only

Weighted Only

	Type	Value	Evaluate Target
MX		5 alt2.aspmx.l.google.com.	
		5 alt1.aspmx.l.google.com.	
		1 aspmx.l.google.com.	
		10 alt3.aspmx.l.google.com.	
		10 alt4.aspmx.l.google.com.	
NS		ns-981.awsdns-58.net.	
		ns-1878.awsdns-42.co.uk.	
		ns-234.awsdns-29.com.	
		ns-1216.awsdns-24.org.	
SOA		ns-234.awsdns-29.com. awsdns-hostmaster.amaz	
		SPF	
TXT		"v=spf1 a include:_spf.google.com include:mailgun.c	
		"google-site-verification=F5uiDwad1Q9LPr3OVbCFz	
TXT		"google-site-verification=mMwwHnaPRLJDaHoZccR	
		ay.io.	
CNAME		d7grb542oi5craoroqxp3vg74rzp3ad.dkim.amazons	
CNAME		dsis3d3j2heuaka2tudc64uguqygmwww.dkim.amazor	
CNAME		lb3ov6upd2ywwztbjdnutxhnuqf3tg7.dkim.amazonse	
TXT		"k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GN/	

Create Record Set

Name: demo.ridgway.io.

Type: A - IPv4 address

Aliases: Yes No

Alias Target: d2tjwqc79ixn9u.cloudfront.net

Alias Hosted Zone ID: Z2FDTNATAQYW2

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d1111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-2.elb.amazonaws.com
- S3 website endpoint: s3-website-us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com
- VPC endpoint: example.us-east-2.vpc.amazonaws.com
- API Gateway custom regional API: d-abode12345.execute-api.us-west-2.amazonaws.com

Routing Policy: Simple

Route 53 responds to queries based only on the values in this record.

Evaluate Target Health: Yes No

Create

Feedback

English (US)

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Distribution Settings

Price Class Use All Edge Locations (Best Performance) 

AWS WAF Web ACL None ⓘ

Alternate Domain Names demo.ridgway.io ⓘ

(CNAMEs)

SSL Certificate ☐ Default CloudFront Certificate (*.cloudfront.net)

Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as <https://d111111abcdef8.cloudfront.net/logo.jpg>).

Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.

☒ Custom SSL Certificate (example.com):

Choose this option if you want your users to access your content by using an alternate domain name, such as <https://www.example.com/logo.jpg>.

You can use a certificate stored in AWS Certificate Manager (ACM) in the US East (N. Virginia) Region, or you can use a certificate stored in IAM.

demo.ridgway.io (158e8652-67fe-4ae1-9...

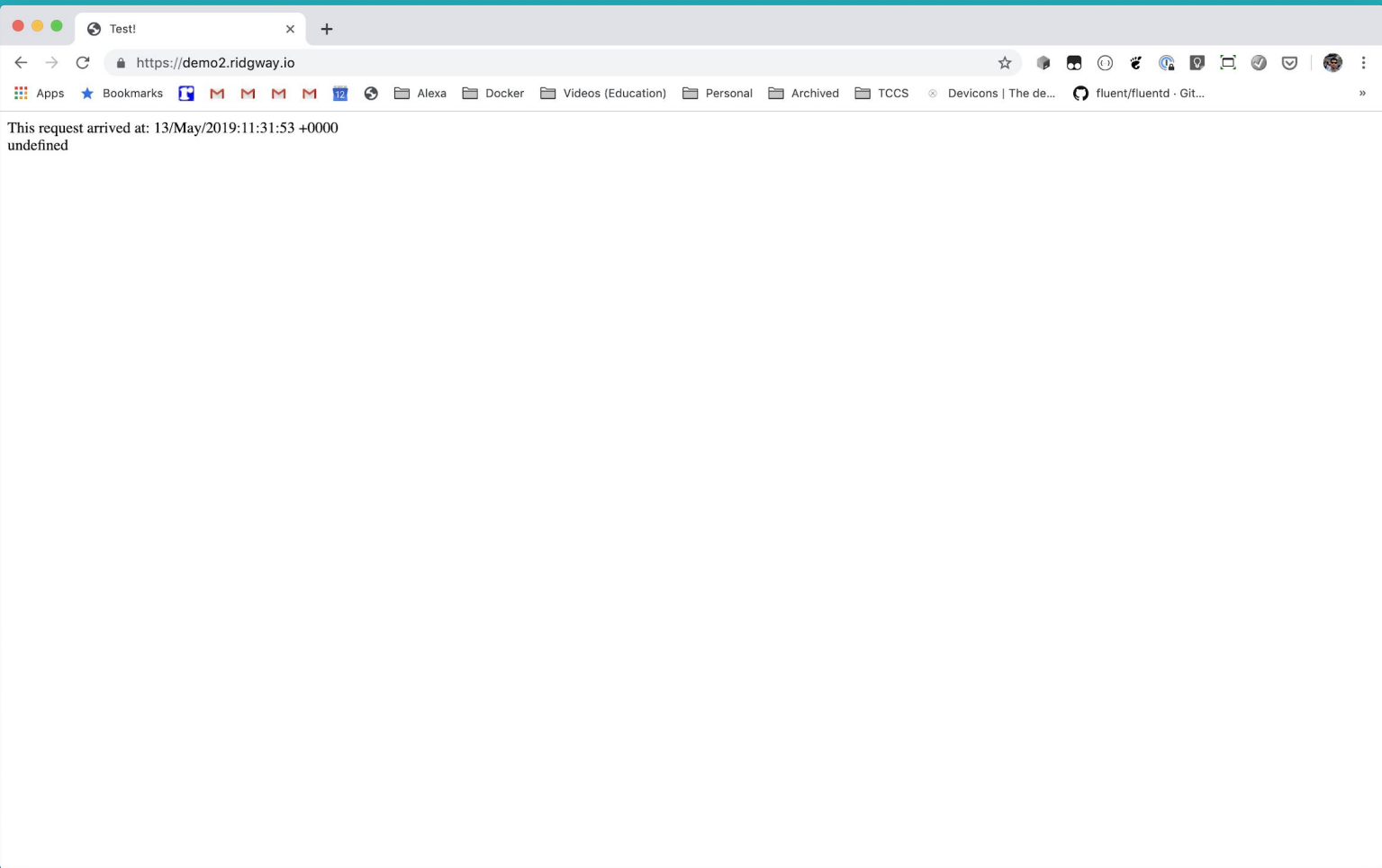
[Request or Import a Certificate with ACM](#)

[Learn more](#) about using custom SSL/TLS certificates with CloudFront.
[Learn more](#) about using ACM.

Custom SSL Client Support [Clients that Support Server Name Indication \(SNI\) - \(Recommended\)](#)

CloudFront serves your content over HTTPS to clients that support SNI. SNI is supported by browsers and clients released after 2010. There is no additional charge for this option.

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Links

- <https://github.com/the-curve-consulting/serverless-ruby-basic>
- <https://github.com/the-curve-consulting/serverless-ruby-with-gems>
- <https://github.com/the-curve-consulting/serverless-ruby-sinatra>

- <https://thecurveconsulting.io>

- <https://www.jamesridgway.co.uk/>

- <https://blockdev.io>
- <https://www.paulridgway.co.uk/>

