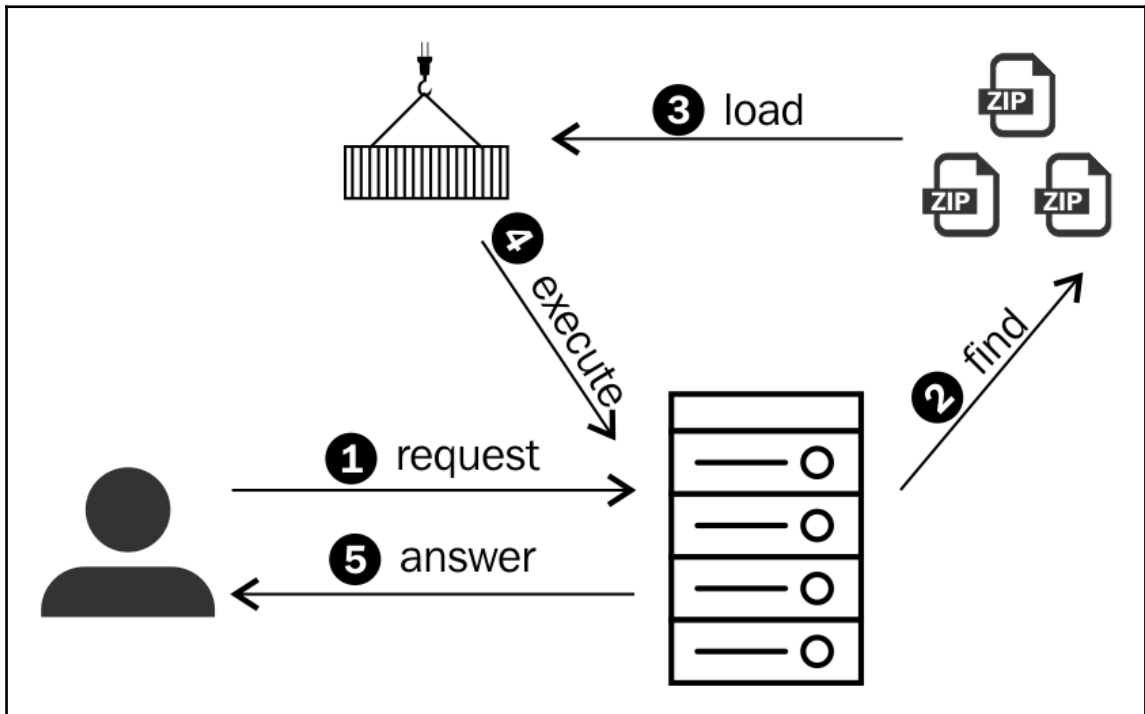
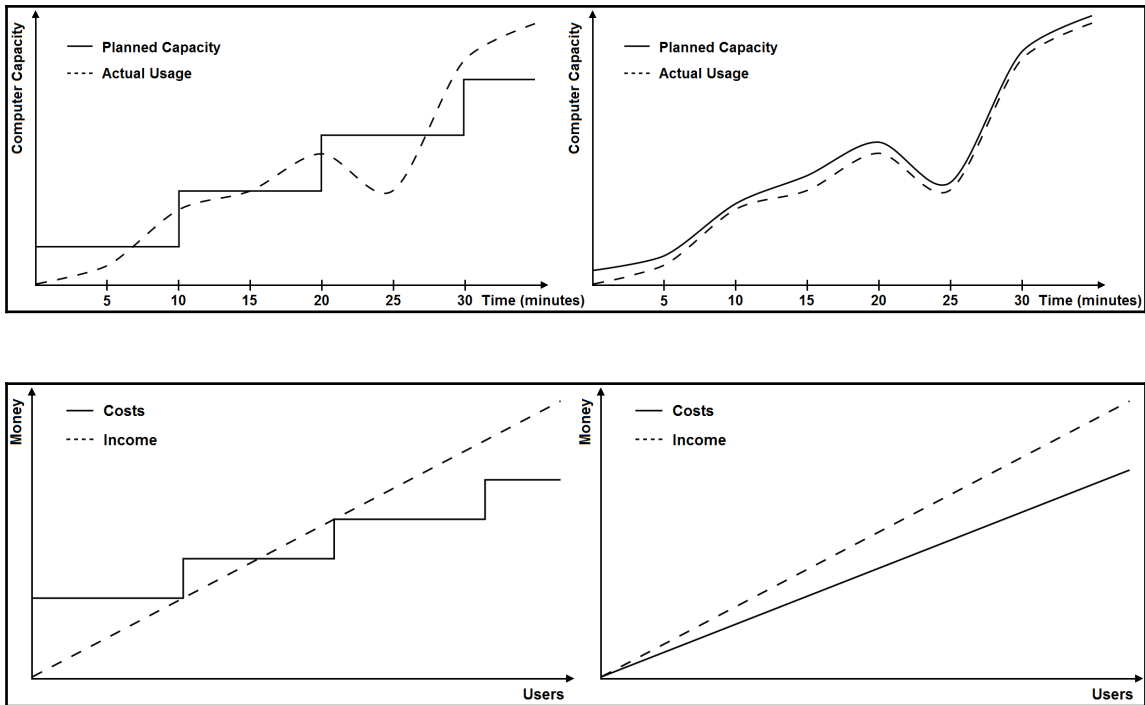


1

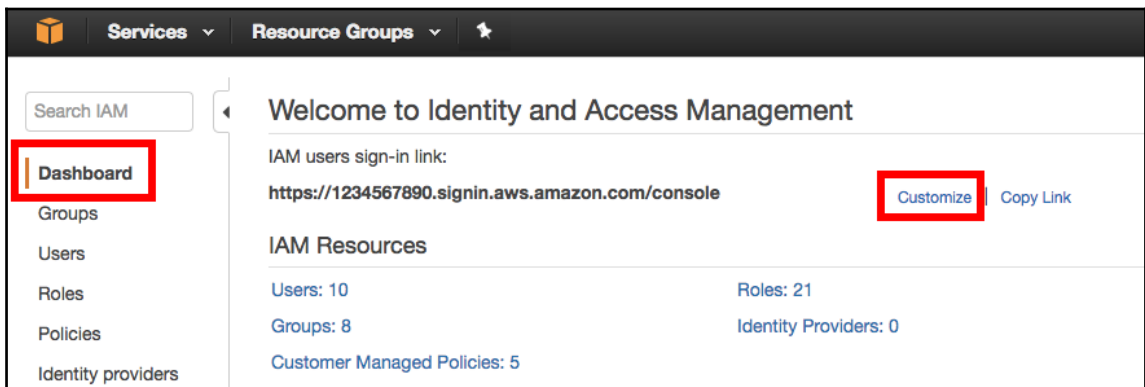
Graphic bundle

Chapter 1: Understanding the Serverless Model





Chapter 2: Getting Started with AWS



Services

Resource Groups

Diego ZanonGlobalSupport

Search IAM

Add user

Delete user

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

Find users by username or access key

Showing 8 results

<input type="checkbox"/>	User name	Groups	Password	Last sign-in	Access keys	Creation time	
<input type="checkbox"/>	user_1	1		N/A	1 active	2015-12-26 18:02 UTC-0300	✕
<input type="checkbox"/>	user_2	1		N/A	1 active	2015-12-27 22:35 UTC-0300	✕
<input type="checkbox"/>	user_3	1		N/A	1 active	2016-08-22 00:28 UTC-0300	✕
<input type="checkbox"/>	user_4	1		N/A	1 active	2016-08-23 18:28 UTC-0300	✕
<input type="checkbox"/>	user_5	1		N/A	1 active	2016-08-23 18:28 UTC-0300	✕
<input type="checkbox"/>	user_6	1		N/A	1 active	2016-11-29 21:53 UTC-0300	✕
<input type="checkbox"/>	user_7	1		N/A	1 active	2017-04-24 20:22 UTC-0300	✕
<input type="checkbox"/>	user_8	1		N/A	1 active	2017-04-24 20:56 UTC-0300	✕

ServicesResource Groups

Diego ZanonGlobalSupport

Add user

1

2

3

4

Details

Permissions

Review

Complete

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

s3_user

+ Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

☒ Programmatic access



☐ AWS Management Console access


Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.

Enables a password that allows users to sign-in to the AWS Management Console.

* Required

CancelNext: Permissions

 **Services** ▾ **Resource Groups** ▾ 

 **Diego Zanon** ▾ **Global** ▾ **Support** ▾

1

2

3

4

Details

Permissions

Review

Complete


Add user to group


Copy permissions from existing user


Attach existing policies directly

Create group

 Refresh

Showing 7 results

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. [Learn more](#)

Group name

s3_fullaccess



Create policy

Refresh

Filter: Policy type

AmazonS3

Showing 2 results

	Policy name	Type	Attachments	Description
<input checked="" type="checkbox"/>	 AmazonS3FullAccess	AWS managed	0	Provides full access to all buckets via th...
<input type="checkbox"/>	 AmazonS3ReadOnly...	AWS managed	0	Provides read only access to all bucket...

Cancel

Create group

Services

Resource Groups

Diego Zanon

Global

Support

Add user

1

2

3

4

Details

Permissions

Review

Complete

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name

s3_user

AWS access type

Programmatic access - with an access key

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	s3_fullaccess

Cancel

Previous

Create user

Services

Resource Groups

Diego Zanon

Global

Support

Add user

1

2

3

4

Details

Permissions

Review

Complete

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://1234567890.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key
▶	✓ s3_user	ABCDEFGHIJKLMNQRST	***** Show

Close

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

☐ Programmatic access
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☒ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*

☒ Autogenerated password
☐ Custom password

Require password reset

☒ User must create a new password at next sign-in
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

Services ▾ Resource Groups ▾ ⌵

Diego Zanon ▾ Global ▾ Support ▾

Add user

1

2

3

4

Details

Permissions

Review

Complete

✓ Success

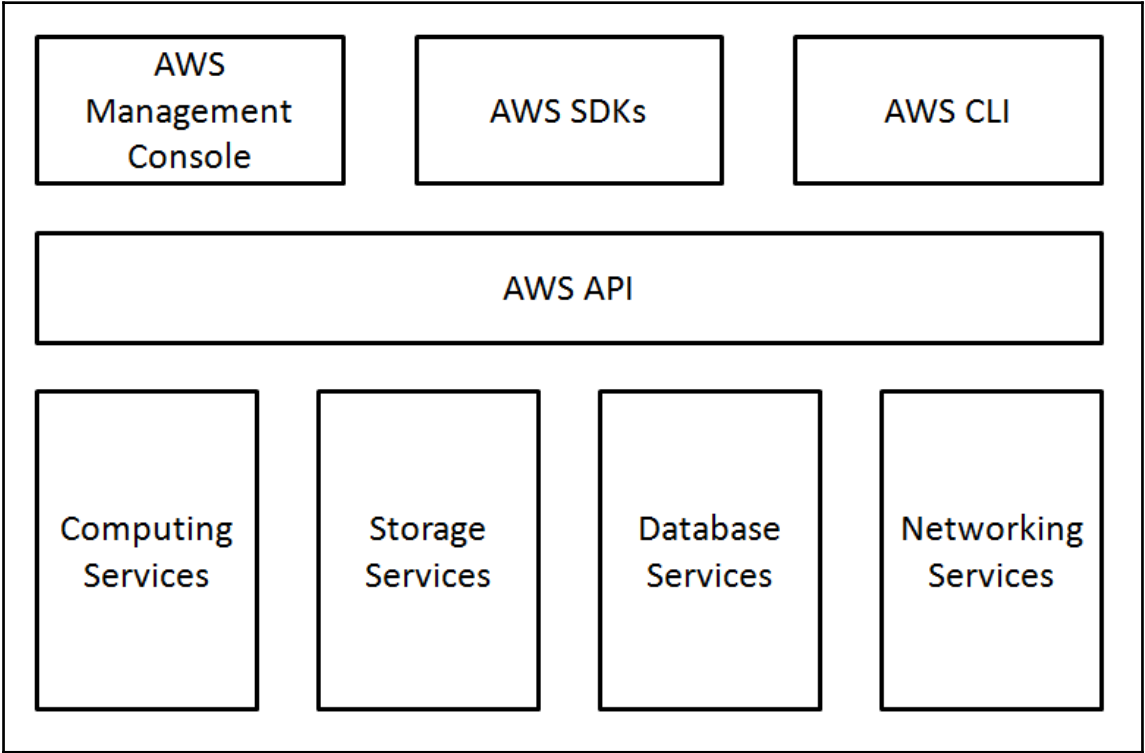
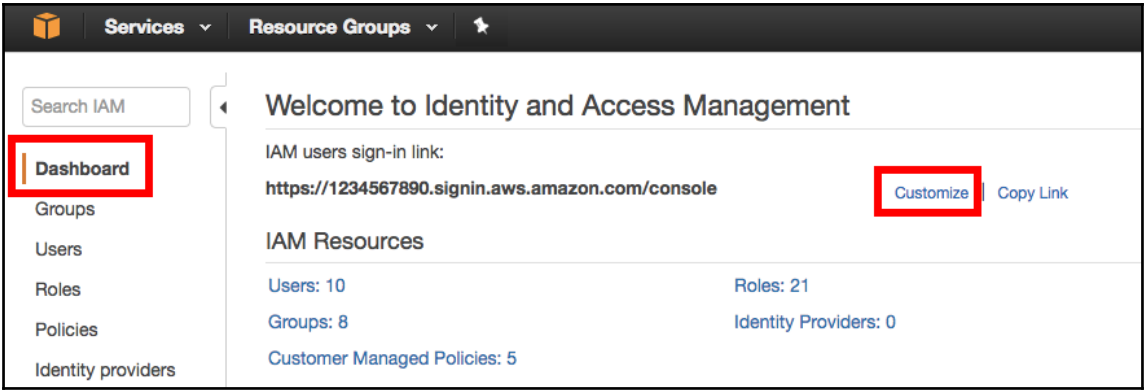
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

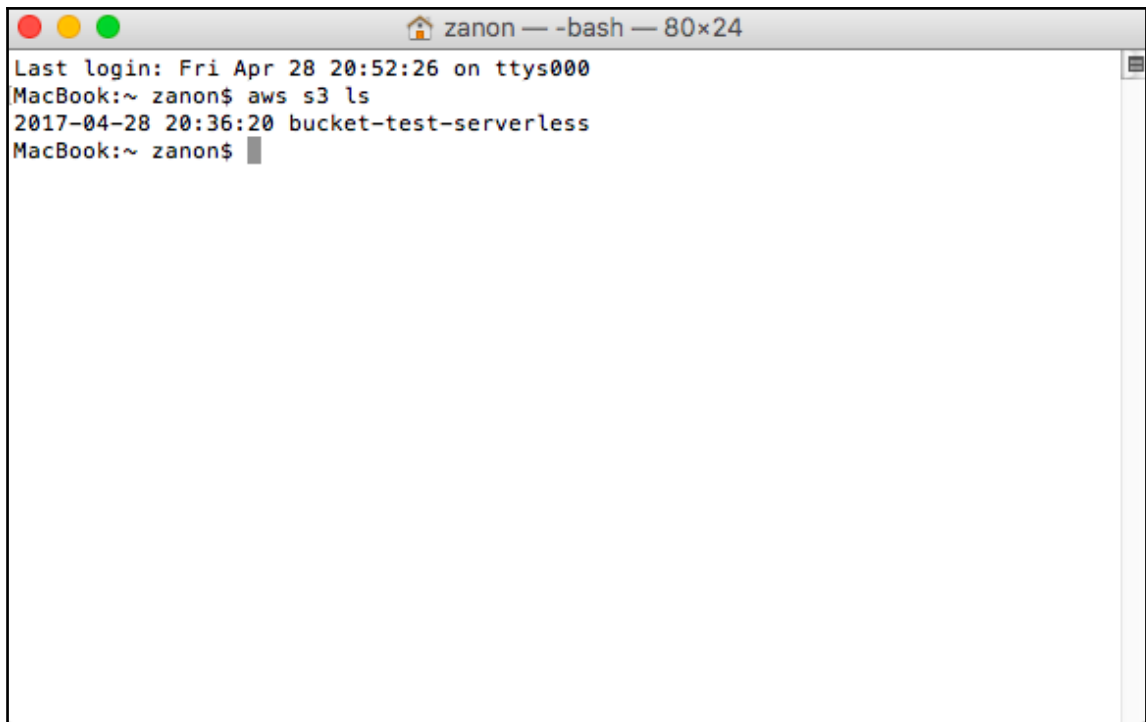
Users with AWS Management Console access can sign-in at <https://1234567890.signin.aws.amazon.com/console>

Download .csv

	User	Password	Email login instructions
▶	✓ console_user	***** Show	Send email ↗

Close






A terminal window titled 'zanon — -bash — 80x24' with standard macOS window controls (red, yellow, green buttons). The terminal displays the following text:

```
Last login: Fri Apr 28 20:52:26 on ttys000
MacBook:~ zanon$ aws s3 ls
2017-04-28 20:36:20 bucket-test-serverless
MacBook:~ zanon$
```

ServicesResource Groups

Diego ZanonN. VirginiaSupport




AWS Lambda

AWS Lambda lets you run code in response to events, without provisioning or managing servers. Just upload your code and Lambda will take care of everything required to run and scale it with high availability.

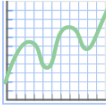
Get Started Now

[Learn more about AWS Lambda](#)




No Servers to Manage

AWS Lambda automatically runs your code without requiring you to provision or manage servers. Just write the code and upload it to Lambda.



Continuous Scaling

AWS Lambda automatically scales your application by running code in response to each trigger. Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload.



Subsecond Metering

With AWS Lambda, you are charged for every 100ms your code executes and the number of times your code is triggered. You don't pay anything when your code isn't running.

ServicesResource Groups

Diego ZanonN. VirginiaSupport

Lambda > New function

Select blueprint

Configure triggers

Configure function

Review

Select blueprint

Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that best aligns with your desired scenario and customize as needed, or skip this step if you want to author a Lambda function and configure an event source separately. Except where otherwise noted, blueprints are licensed under CC0.

Welcome to AWS Lambda! You can get started on creating your first Lambda function by choosing one of the blueprints below.

Select runtimeFilter

<< Viewing 1-9 of 89 >>

Blank Function

Configure your function from scratch. Define the trigger and deploy your code by stepping through our wizard.

custom

kinesis-firehose-syslog-to-json

An Amazon Kinesis Firehose stream processor that converts input records from RFC3164 Syslog format to JSON.

nodejs · kinesis-firehose

alex-skill-kit-sdk-factskill

Demonstrate a basic fact skill built with the ASK NodeJS SDK

nodejs · alexa

batch-get-job-python27

Returns the current status of an AWS Batch Job.

python2.7 · batch

kinesis-firehose-apachelog-to-...

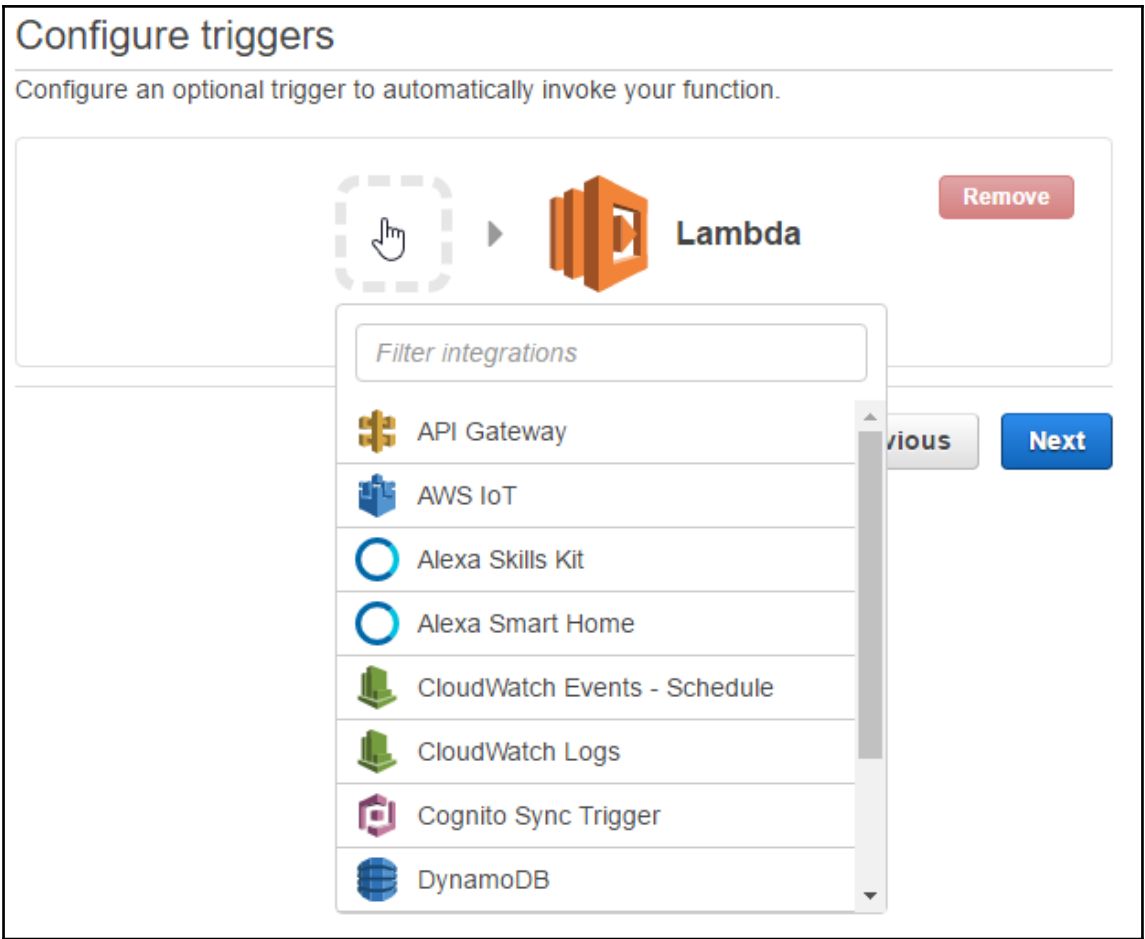
An Amazon Kinesis Firehose stream processor that converts input records from Apache Common Log format to

python2.7 · kinesis-firehose

cloudfront-modify-response-h...

Blueprint for modifying CloudFront response header implemented in NodeJS.


nodejs · cloudfront · response header




Configure triggers

Configure an optional trigger to automatically invoke your function.

S3



►



Lambda

Remove

Bucket

my-bucket-name

▼

i

Event type

Object Created (All)

▼

i

Prefix

logs/

i

Suffix

i

Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this bucket. [Learn more](#) about the Lambda permissions model.

Enable trigger

☒

i

Cancel

Previous

Next

Configure function

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

Name*

processLog

Description

Process S3 files and notify SNS on errors

Runtime*

Node.js 6.10

▼

[12]

Lambda function handler and role

Handler* ⓘ

Role*

Create a custom role ⓘ

Existing role*

Choose an existing role ⓘ

Create new role from template(s) ⓘ

Create a custom role

▼ Hide Details

Role Summary ⓘ

Role Lambda execution role permissions

Description

IAM Role

Role Name

▼ Hide Policy Document

[Edit](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ]
    }
  ],
}
```

[Don't Allow](#) [Allow](#)

▼ 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

Create new API

In Amazon API Gateway, an API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

☒ New API

☐ Clone from existing API

☐ Import from Swagger

☐ Example API

Name and description

Choose a friendly name and description for your API.

API name*

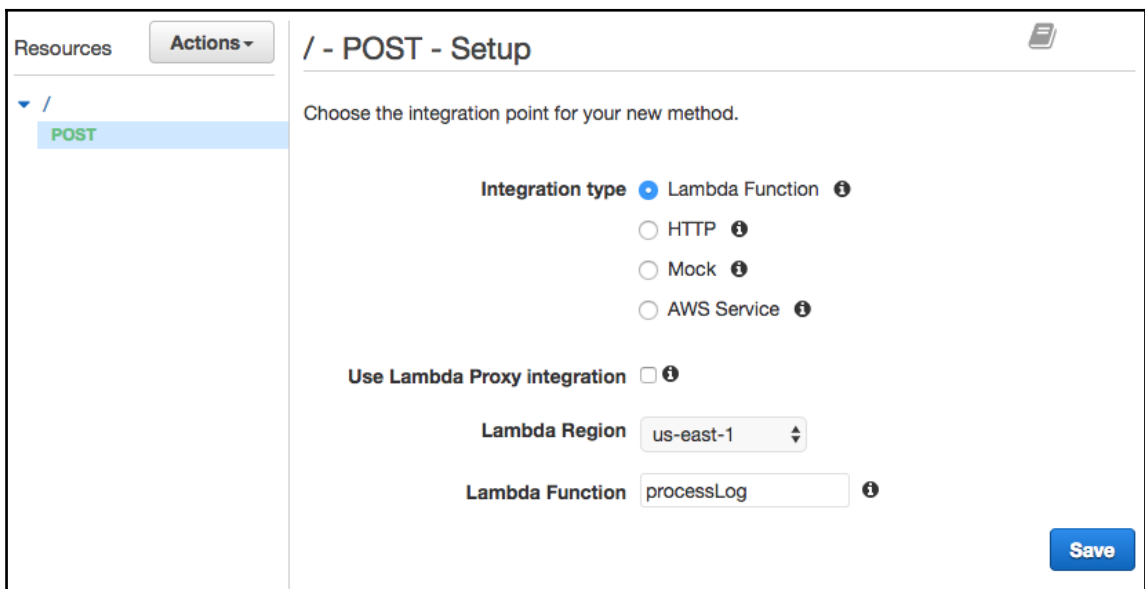
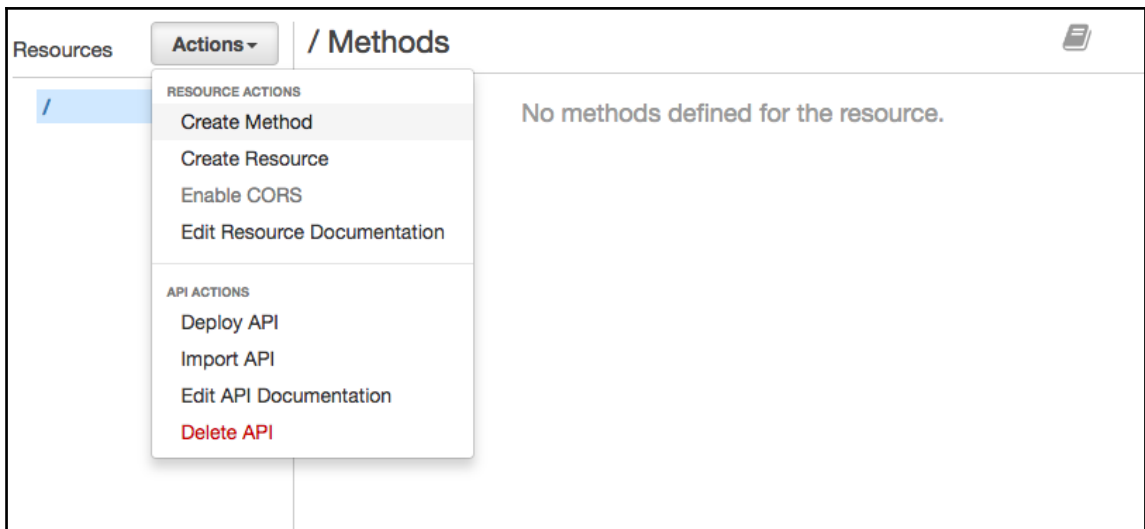
log-processor

Description

Process S3 logs to find errors

* Required

Create API



Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Deployment stage

[New Stage]

Stage name*

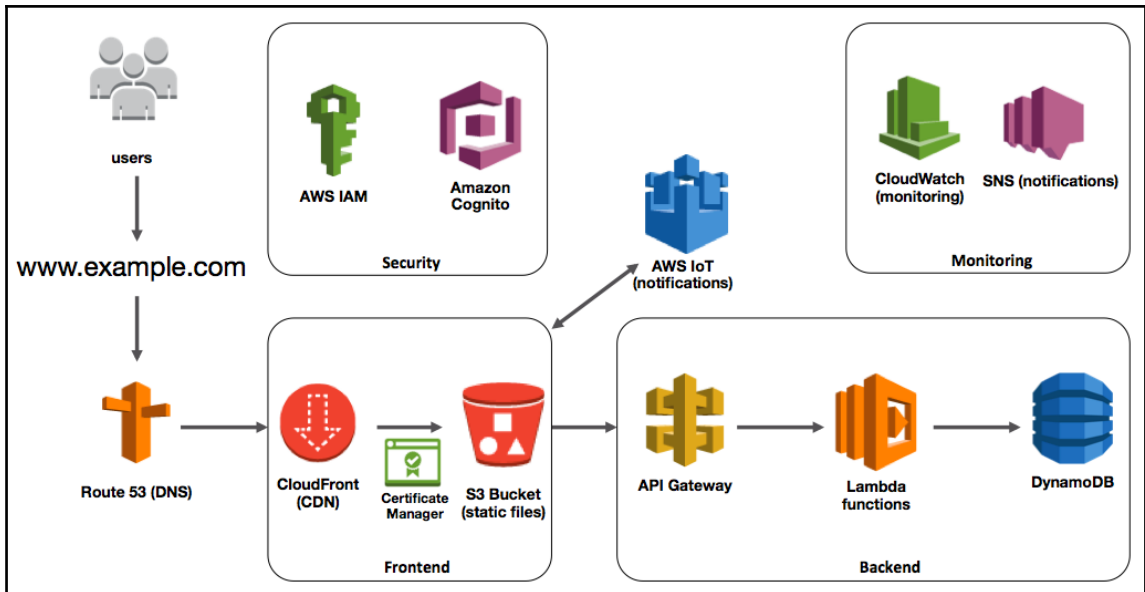
dev

Stage description

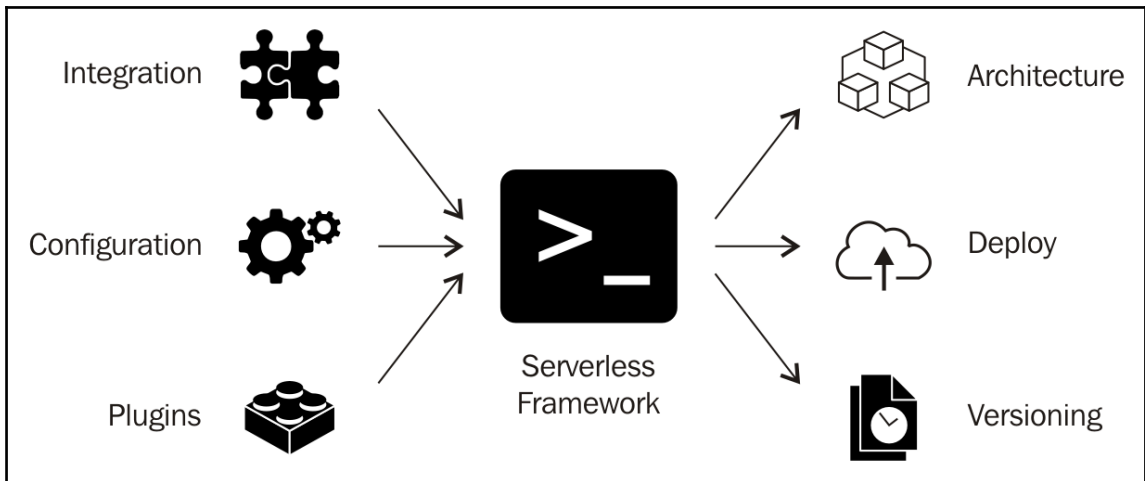
Deployment description

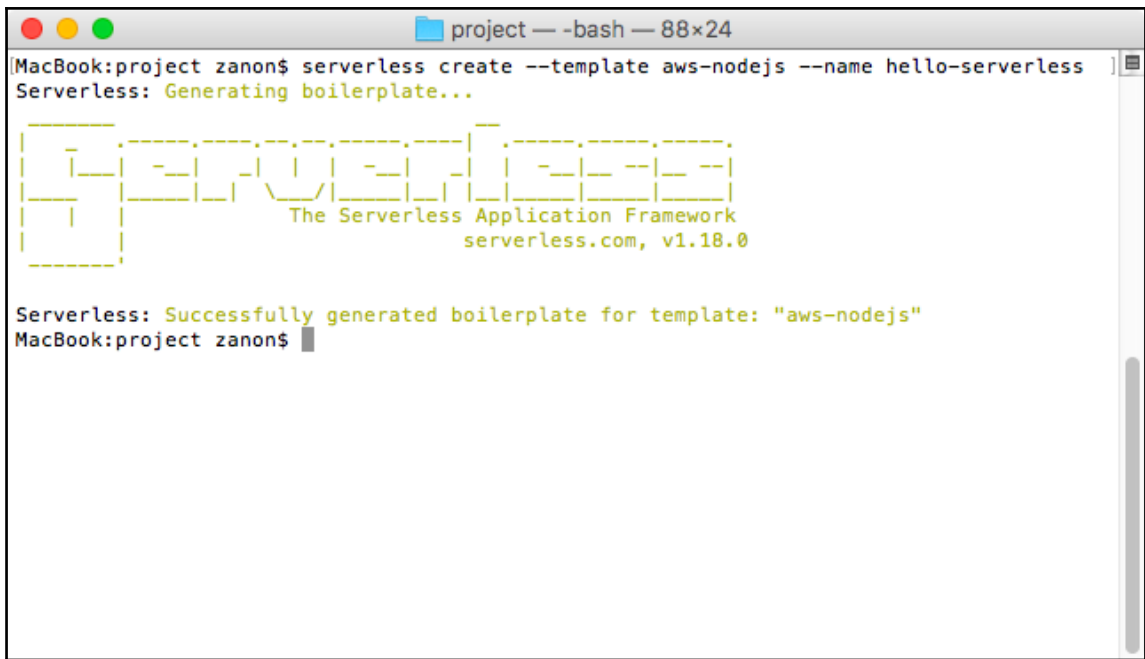
Cancel

Deploy



Chapter 3: Using the Serverless Framework



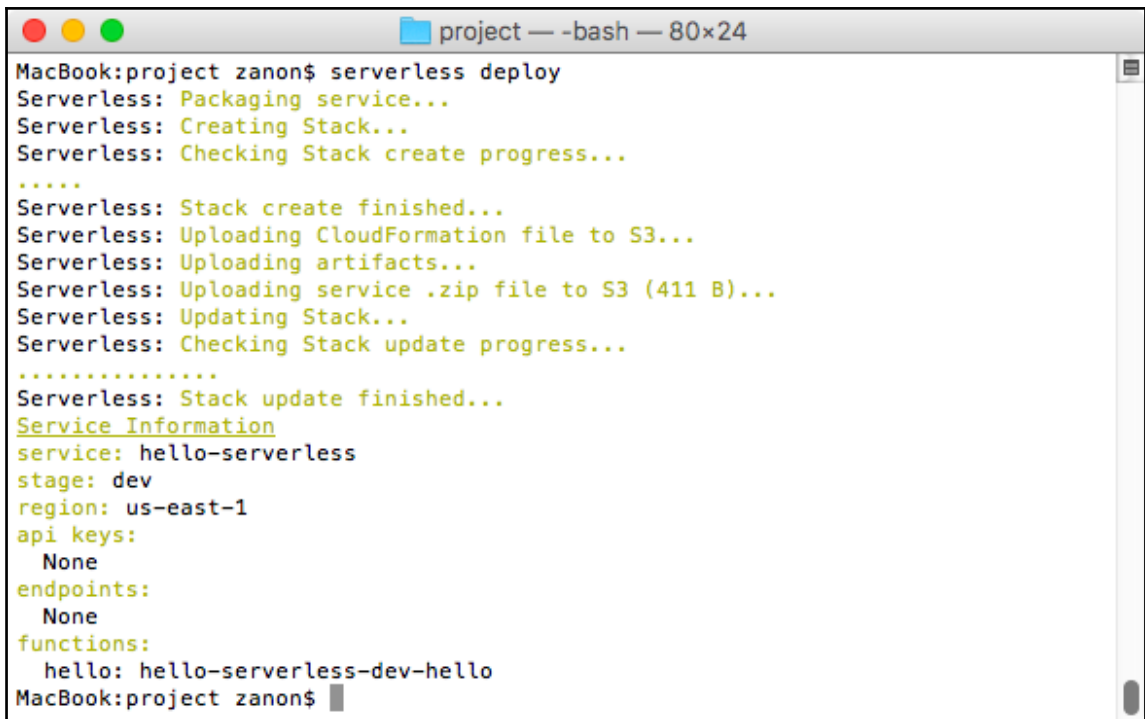


A terminal window titled "project — -bash — 88x24" is shown. The prompt is "MacBook:project zanon\$". The command "serverless create --template aws-nodejs --name hello-serverless" has been entered. The output shows "Serverless: Generating boilerplate..." followed by a yellow ASCII art logo for "serverless". The logo consists of the word "serverless" in a stylized, blocky font, with "The Serverless Application Framework" and "serverless.com, v1.18.0" written below it. The final output is "Serverless: Successfully generated boilerplate for template: "aws-nodejs"". The prompt is now "MacBook:project zanon\$".

```
MacBook:project zanon$ serverless create --template aws-nodejs --name hello-serverless
Serverless: Generating boilerplate...

serverless
The Serverless Application Framework
serverless.com, v1.18.0

Serverless: Successfully generated boilerplate for template: "aws-nodejs"
MacBook:project zanon$
```

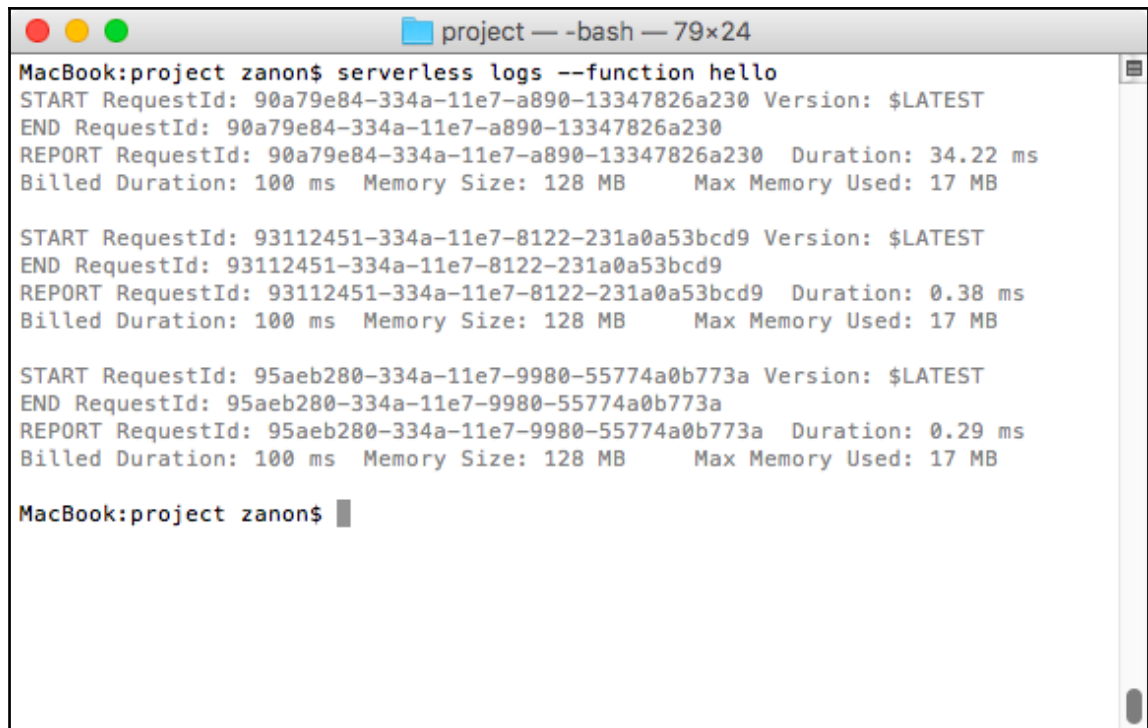


```
MacBook:project zanon$ serverless deploy
Serverless: Packaging service...
Serverless: Creating Stack...
Serverless: Checking Stack create progress...
.....
Serverless: Stack create finished...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service .zip file to S3 (411 B)...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.....
Serverless: Stack update finished...
Service Information
service: hello-serverless
stage: dev
region: us-east-1
api keys:
  None
endpoints:
  None
functions:
  hello: hello-serverless-dev-hello
MacBook:project zanon$
```



A terminal window titled "project — -bash — 80x24" is shown. The prompt is "MacBook:project zanon\$". The command "serverless invoke --function hello --path event.json" has been executed. The output is a JSON object: {"statusCode": 200, "body": "{\"message\": \"Hello, Serverless!\"}"}. The prompt is now "MacBook:project zanon\$".

```
MacBook:project zanon$ serverless invoke --function hello --path event.json
{
  "statusCode": 200,
  "body": "{\"message\": \"Hello, Serverless!\"}"
}
MacBook:project zanon$
```

A terminal window titled "project — -bash — 79x24" with a grey title bar and three colored window control buttons (red, yellow, green) on the left. The terminal content shows the output of the command "serverless logs --function hello". It displays three log entries for different request IDs, each with "START", "END", and "REPORT" lines. The first entry has a duration of 34.22 ms, while the others are 0.38 ms and 0.29 ms. All entries show a billed duration of 100 ms, a memory size of 128 MB, and a max memory used of 17 MB. The prompt "MacBook:project zanon\$" is visible at the bottom.

```
MacBook:project zanon$ serverless logs --function hello
START RequestId: 90a79e84-334a-11e7-a890-13347826a230 Version: $LATEST
END RequestId: 90a79e84-334a-11e7-a890-13347826a230
REPORT RequestId: 90a79e84-334a-11e7-a890-13347826a230  Duration: 34.22 ms
Billed Duration: 100 ms  Memory Size: 128 MB      Max Memory Used: 17 MB

START RequestId: 93112451-334a-11e7-8122-231a0a53bcd9 Version: $LATEST
END RequestId: 93112451-334a-11e7-8122-231a0a53bcd9
REPORT RequestId: 93112451-334a-11e7-8122-231a0a53bcd9  Duration: 0.38 ms
Billed Duration: 100 ms  Memory Size: 128 MB      Max Memory Used: 17 MB

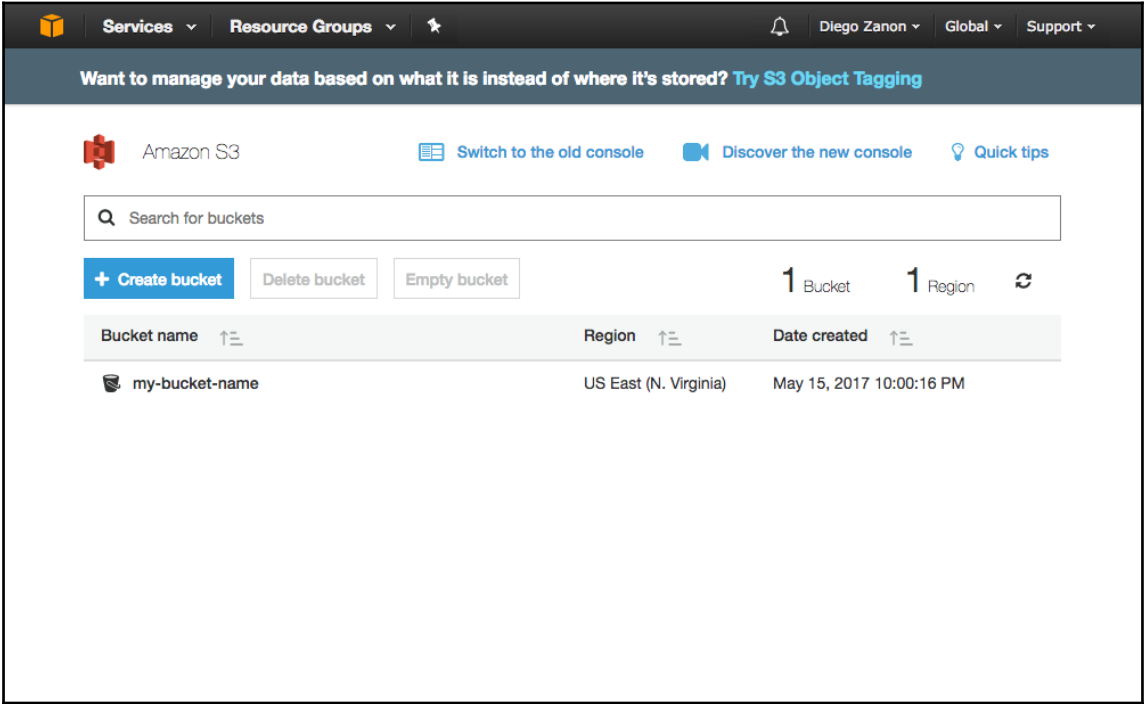
START RequestId: 95aeb280-334a-11e7-9980-55774a0b773a Version: $LATEST
END RequestId: 95aeb280-334a-11e7-9980-55774a0b773a
REPORT RequestId: 95aeb280-334a-11e7-9980-55774a0b773a  Duration: 0.29 ms
Billed Duration: 100 ms  Memory Size: 128 MB      Max Memory Used: 17 MB

MacBook:project zanon$
```

```
project — -bash — 86×24
MacBook:project zanon$ serverless deploy
Serverless: Packaging service...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service .zip file to S3 (444 B)...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.....
Serverless: Stack update finished...
Service Information
service: hello-serverless
stage: dev
region: us-east-1
api keys:
  None
endpoints:
  GET - https://dxch4zvl5j.execute-api.us-east-1.amazonaws.com/dev/my-service/resource
functions:
  hello: hello-serverless-dev-hello
MacBook:project zanon$
```

```
project — -bash — 86×24
MacBook:project zanon$ serverless info
Service Information
service: hello-serverless
stage: dev
region: us-east-1
api keys:
  None
endpoints:
  GET - https://dxch4zvl5j.execute-api.us-east-1.amazonaws.com/dev/my-service/resource
functions:
  hello: hello-serverless-dev-hello
MacBook:project zanon$
```

Chapter 4: Hosting the Website



Create bucket

1

Name and region

2

Set properties


3

Set permissions

4


Review

Name and region

Bucket name 


example.com

Region

US East (N. Virginia) 

Copy settings from an existing bucket

Select bucket (optional)

1 Bucket 

Create

Cancel

Next

Create bucket

Name and region

2

Set properties

3

Set permissions

4

Review

Versioning

Keep multiple versions of an object in the same bucket.

Learn more

Disabled

Logging

Set up access log records that provide details about access requests.

Learn more

Disabled

Tags

Use tags to track your cost against projects or other criteria.

Learn more

0 Tags

Previous

Next

Create bucket

✓

Name and region

✓

Set properties

3

Set permissions

4

Review

▸ Manage users

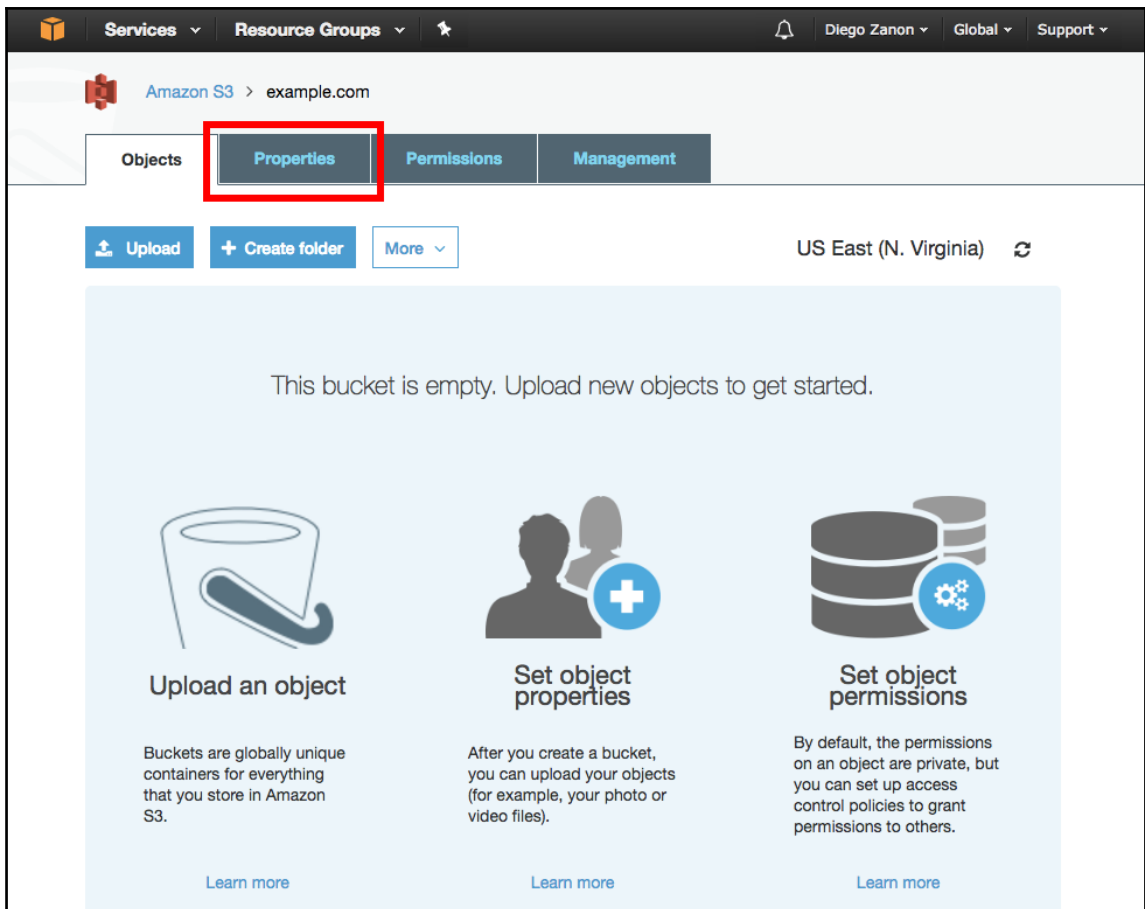
▼ Manage group permissions

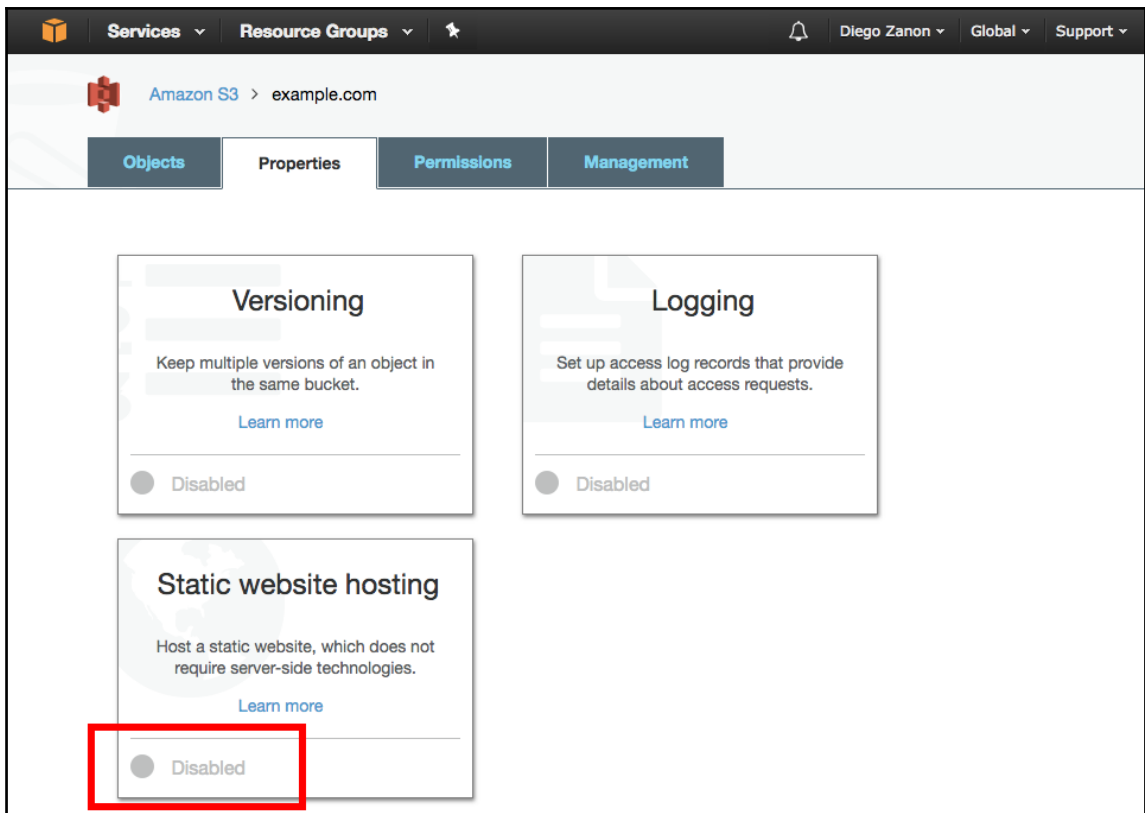
Group	Objects	Object permissions
Any authenticated AWS user	<input type="checkbox"/> Read <input type="checkbox"/> Write	<input type="checkbox"/> Read <input type="checkbox"/> Write
Everyone	<input checked="" type="checkbox"/> Read <input type="checkbox"/> Write	<input type="checkbox"/> Read <input type="checkbox"/> Write

▸ Manage system permissions

Previous

Next





Static website hosting

Endpoint : http://example.com.s3-website-us-east-1.amazonaws.com

☒ Use this bucket to host a website [Learn more](#)

Index document [i](#)

index.html

Error document [i](#)

error.html

Redirection rules (optional) [i](#)

<RoutingRules>
<RoutingRule>
<Condition>
<HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedE
</Condition>
<Redirect>
<ReplaceKeyWith>not-found.html</ReplaceKeyWith>
</Redirect>
</RoutingRule>
</RoutingRules>

☐ Redirect requests [i](#) [Learn more](#)

☐ Disable website hosting

Cancel

Save

[29]

Static website hosting

Endpoint : <http://www.example.com.s3-website-us-east-1.amazonaws.com>

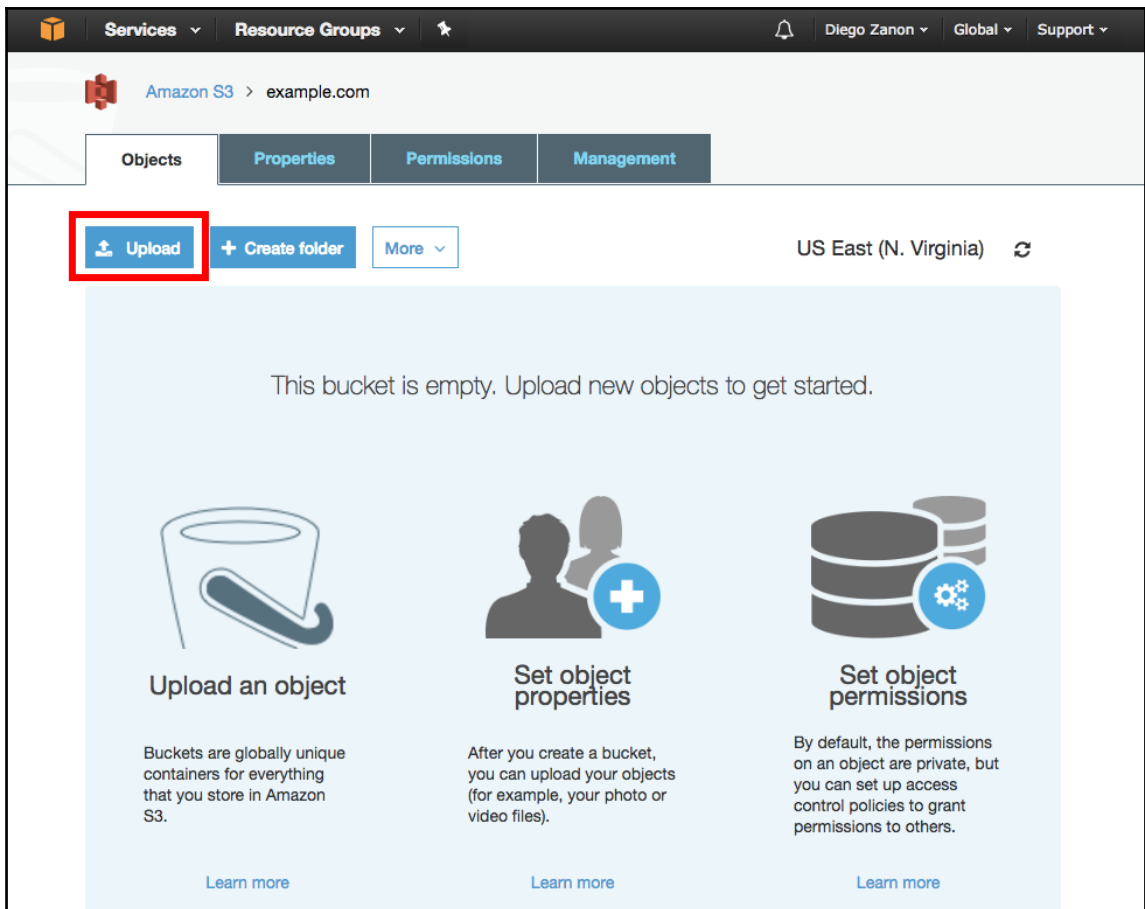
☐ Use this bucket to host a website [Learn more](#)

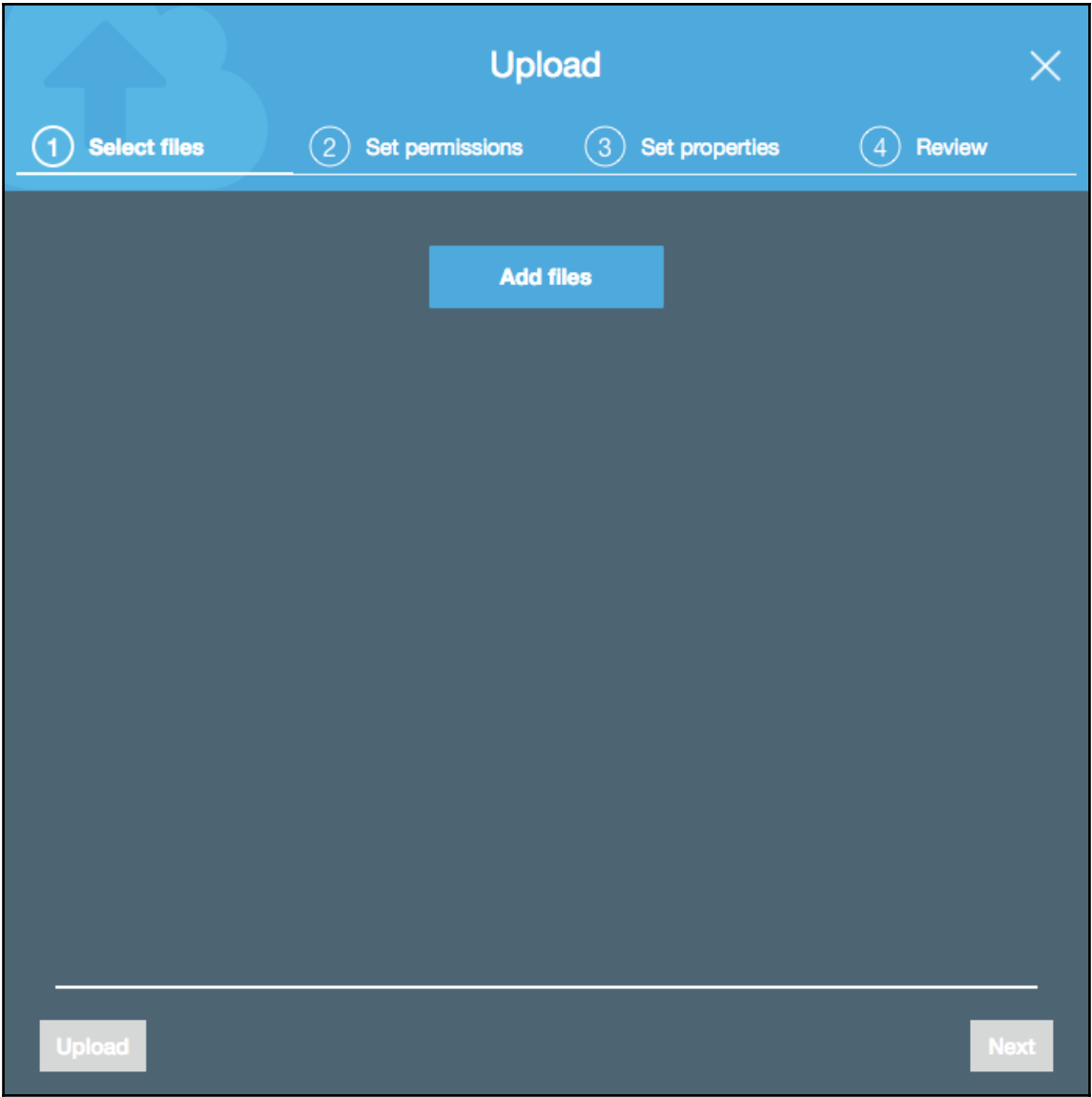
☒ Redirect requests [Learn more](#)


Target bucket or domain

Protocol


☐ Disable website hosting











Upload



 Select files

 Set permissions




 Set properties

 Review

1 Files **Size:** 124.0 B **Target path:** serverless-store.zanon.io

▸ Manage users

▾ Manage group permissions

Group 	Objects 	Object permissions 
Any authenticated AWS user	<input type="checkbox"/> Read <input type="checkbox"/> Write	<input type="checkbox"/> Read <input type="checkbox"/> Write
Everyone	<input checked="" type="checkbox"/> Read <input type="checkbox"/> Write	<input type="checkbox"/> Read <input type="checkbox"/> Write

Upload

Previous

Next

Upload

✓

Select files

✓

Set permissions

3

Set properties

4

Review

1 Files

Size: 124.0 B

Target path: serverless-store.zanon.io

Storage class

Choose one depending on your use case scenario and performance access requirements.

☒ Standard

☐ Standard-IA

☐ Reduced redundancy

Encryption

Protect data at rest by using Amazon S3 master-key or by using AWS KMS master-key.

☒ None

☐ Amazon S3 master-key

☐ AWS KMS master-key

Metadata

Metadata is a set of name-value pairs. You cannot modify object metadata after it is uploaded.

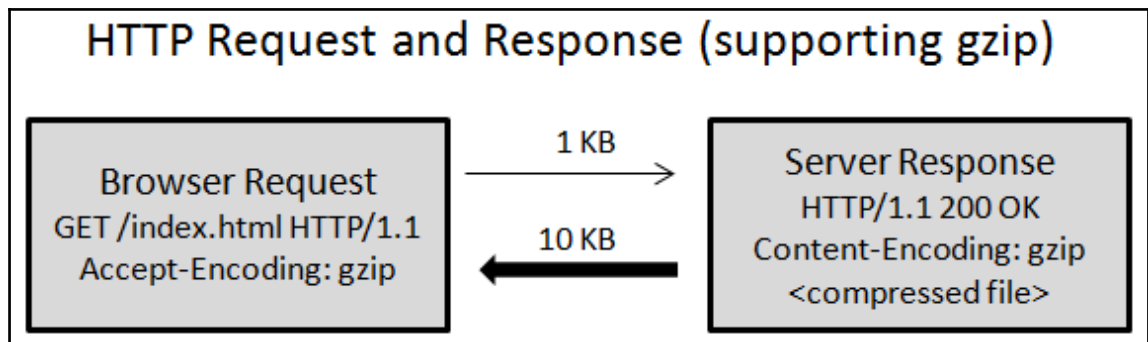
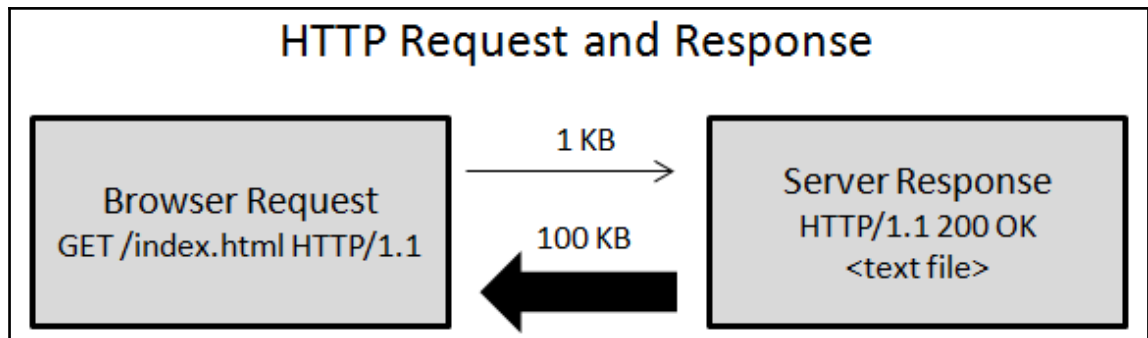
Header

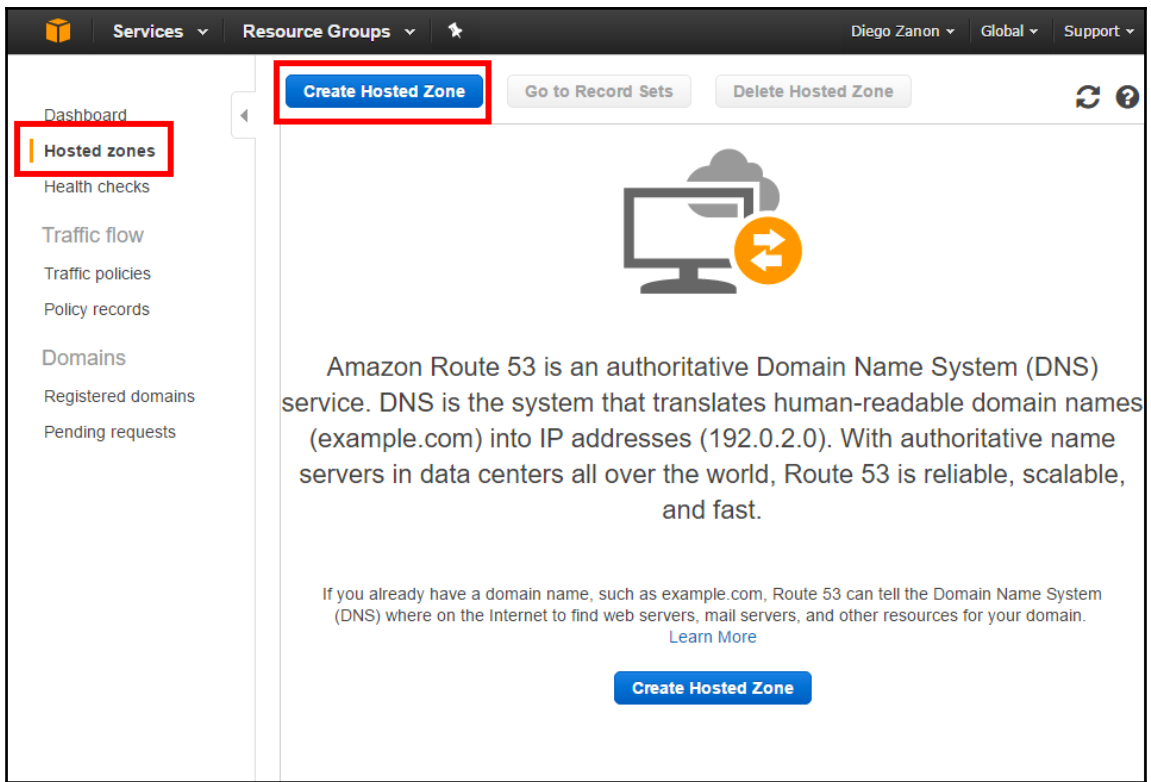
Value

Upload

Previous

Next





Create Hosted Zone

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain Name:

example.com

Comment:

Type:

Public Hosted Zone

A public hosted zone determines how traffic is routed on the Internet.

<input type="checkbox"/>	example.com	NS	ns-2486.awsdns-52.co.uk. ns-1349.awsdns-34.org. ns-59.awsdns-09.com. ns-543.awsdns-27.net.
<input type="checkbox"/>	example.com	SOA	ns-2486.awsdns-52.co.uk. awsdns-hostmaster.amaz

Dashboard

Hosted zones

Health checks

Traffic flow

Back to Hosted Zones

Create Record Set

Import Zone File

Record Set Name

Any Type

Aliases Only

Name	Type	Value
------	------	-------

Create Record Set

Name:

Type:

Alias: ☒ Yes ☐ No

Alias Target:

Alias Hosted Zone ID: ABCDEFGHIJKLMN

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

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy:

Route 53 responds to queries based only on the values in this record. [Learn More](#)

Evaluate Target Health: ☐ Yes ☒ No

Create Record Set

Name:

www

example.com

Type:

CNAME – Canonical name

Alias:

☐ Yes ☒ No

TTL (Seconds):

300

1m

5m

1h

1d

Value:

www.example.com.s3-website-us-east-1.amazonaws.com

The domain name that you want to resolve to instead of the value in the Name field.

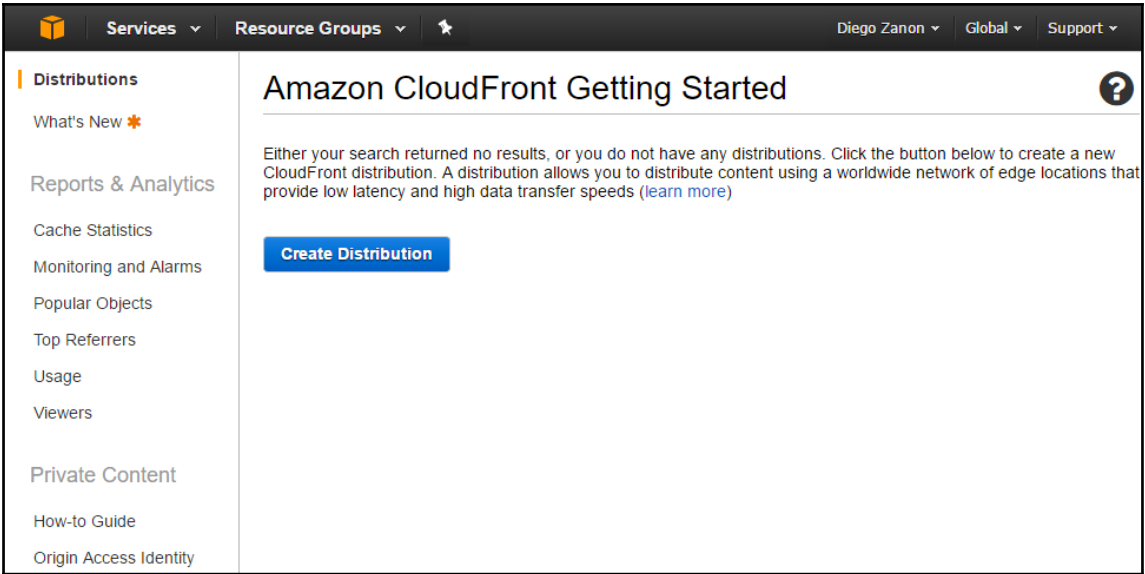
Example:

www.example.com

Routing Policy:

Simple

Route 53 responds to queries based only on the values in this record. [Learn More](#)



Select a delivery method for your content.



Web

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin - either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

Get Started

RTMP

Create an RTMP distribution to speed up distribution of your streaming media files using Adobe Flash Media Server's RTMP protocol. An RTMP distribution allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location. Note the following:

- To create an RTMP distribution, you must store the media files in an Amazon S3 bucket.
- To use CloudFront live streaming, create a web distribution.

Get Started

Cancel

Create Distribution

Origin Settings

Origin Domain Name

example.com.s3-website-us-east-1.ama

Origin Path

Origin ID

S3-Website-example.com.s3-website-us

Origin Custom Headers

Header Name

Default Cache Behavior Settings

Path Pattern

Default (*)

Viewer Protocol Policy

☒ HTTP and HTTPS

☐ Redirect HTTP to HTTPS

☐ HTTPS Only

Allowed HTTP Methods

☐ GET, HEAD

☐ GET, HEAD, OPTIONS

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

Cached HTTP Methods

GET, HEAD (Cached by default)

☒ OPTIONS

Forward Headers

None (Improves Caching)

Object Caching

☒ Use Origin Cache Headers

☐ Customize

Learn More

Minimum TTL

0

i

Maximum TTL

31536000

i

Default TTL

86400

i

Forward Cookies

None (Improves Caching) ▾

i

Query String Forwarding and Caching

None (Improves Caching) ▾

i

Smooth Streaming

☐ Yes

☒ No

i

Restrict Viewer Access
(Use Signed URLs or Signed Cookies)

☐ Yes

☒ No

i

Compress Objects Automatically

☒ Yes

☐ No

i

[Learn More](#)

Lambda Function Associations

Event Type

Lambda Function ARN

▾

i

+

Distribution Settings

Price Class

Use All Edge Locations (Best Performance) ▾

i

AWS WAF Web ACL

None ▾

i

Alternate Domain Names (CNAMEs)

example.com, www.example.com

i

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.

CloudFront Distributions

Create Distribution Distribution Settings Delete Enable Disable

Viewing: Any Delivery Method ▾ Any State ▾

Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last
<input type="checkbox"/> Web	ABCDEFGHIJKLMNO	abcdefghijklmn.cloudfront.net	-	example.com.s	example.com,	Deployed	Enabled	2016

Viewing 1 to 6 of

zohoMailHelp | Forum |

Domain Setup

Verify Domain ✓

Add Users ✓

Create Groups ✓

Configure Email Delivery ✓

SPF/DKIM ✓

Email Migration ✓

Mail Client Configuration ✓

Mobile Access ✓

Welcome to Zoho Mail Suite

You are about to setup business email for your domain **example.com** in Zoho. Follow the instructions here to complete the steps for email setup. You need to start by verifying your domain. [Learn More](#)

Verify the ownership of **EXAMPLE.COM** [Edit](#) | [Verify domain later](#)

Select your domain's DNS Manager (DNS Hosting provider) from the list:

Others..

CNAME Method | [TXT Method](#) | [HTML Method](#)

The zb code for your domain is zb44760758. Create a CNAME for your domain **example.com** in your DNS Provider and point it to **zmverify.zoho.com**:

Name / Host / Alias / CNAME	Value / Points To / Destination
zb44760758	zmverify.zoho.com

Proceed to CNAME Verification

zohoMailHelp | Forum |

Domain Setup

Verify Domain ✓

Add Users ✓

Create Groups ✓

Configure Email Delivery ✓

SPF/DKIM ✓

Email Migration ✓

Mail Client Configuration ✓

Mobile Access ✓

Configure Email Delivery - Start receiving emails

You need to configure the MX records of your domain in the DNS hosting provider (DNS Manager) to start receiving email to the users and groups created. You change the email service provider of the domain, only after this critical step.

MX Records (Mail eXchange) are the special entries in DNS that designate the email-receiving server of your domain. Ensure that you have created the required user accounts and group accounts, before changing the MX.

The MX Records of Zoho are:

Host Name	Address	Priority
@	mx.zoho.com	10
@	mx2.zoho.com	20

You must remove (delete) any other MX records other than the above 2 records. In case you have previous providers MX records, then you may not receive emails in Zoho.

Create Record Set

Name:

Type:

Alias: ☐ Yes ☒ No

TTL (Seconds):

Value:

10 mx.zoho.com

20 mx2.zoho.com

A priority and a domain name that specifies a mail server. Enter multiple values on separate lines.

Format:

[priority] [mail server host name]

Example:

10 mailserver.example.com.

20 mailserver2.example.com.

Routing Policy:

Route 53 responds to queries based only on the values in this record. [Learn More](#)

Services

Resource Groups

Diego Zanon

N. Virginia

Support

Request a certificate

Step 1: Add domain names

Step 2: Review and request

Step 3: Validation

You can use AWS Certificate Manager certificates only with Elastic Load Balancing and Amazon CloudFront. [Learn more.](#)

Add domain names

Type the fully qualified domain name of the site you want to secure with an SSL/TLS certificate (for example, `www.example.com`). Use an asterisk (*) to request a wildcard certificate to protect several sites in the same domain. For example: `*.example.com` protects `www.example.com`, `site.example.com` and `images.example.com`.

Domain name*

example.com

Remove

www.example.com

*

Add another name to this certificate

You can add additional names to this certificate. For example, if you're requesting a certificate for `*www.example.com`, you might want to add the name `*example.com` so that customers can reach your site by either name. [Learn more.](#)

*At least one domain name is required

Cancel

Review and request

[47]

Edit Distribution

Distribution Settings

Price Class Use All Edge Locations (Best Performance) ▼



AWS WAF Web ACL None ▼



Alternate Domain Names (CNAMEs) example.com, www.example.com



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://d1111111abcdef8.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.

example.com (15fcf39b-1f1b-48dd-a142-c25f1c... ▼










Request or Import a Certificate with ACM

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

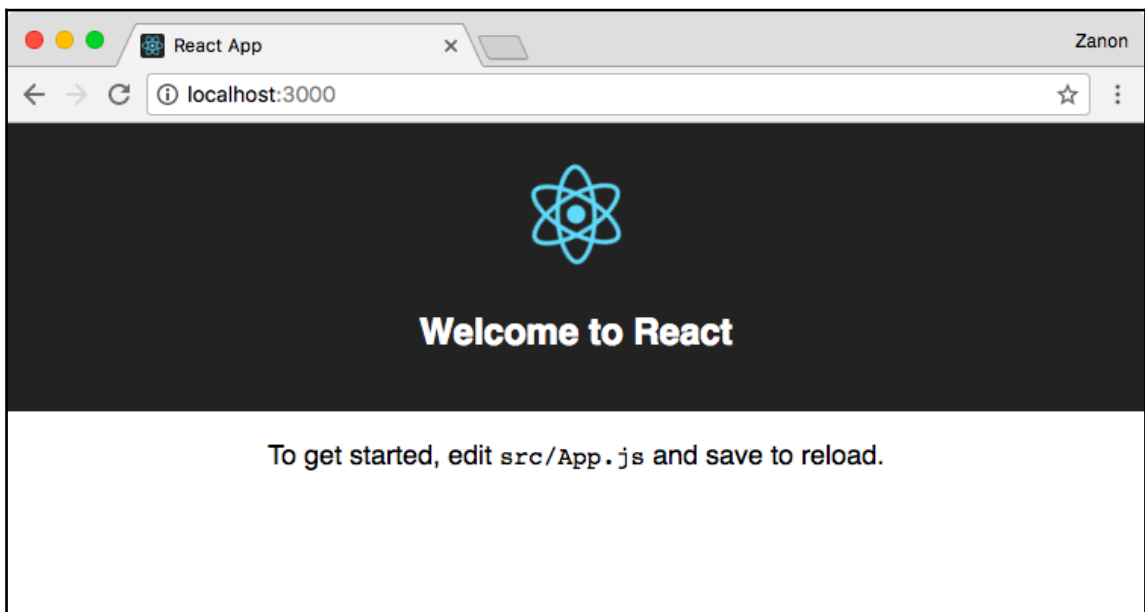
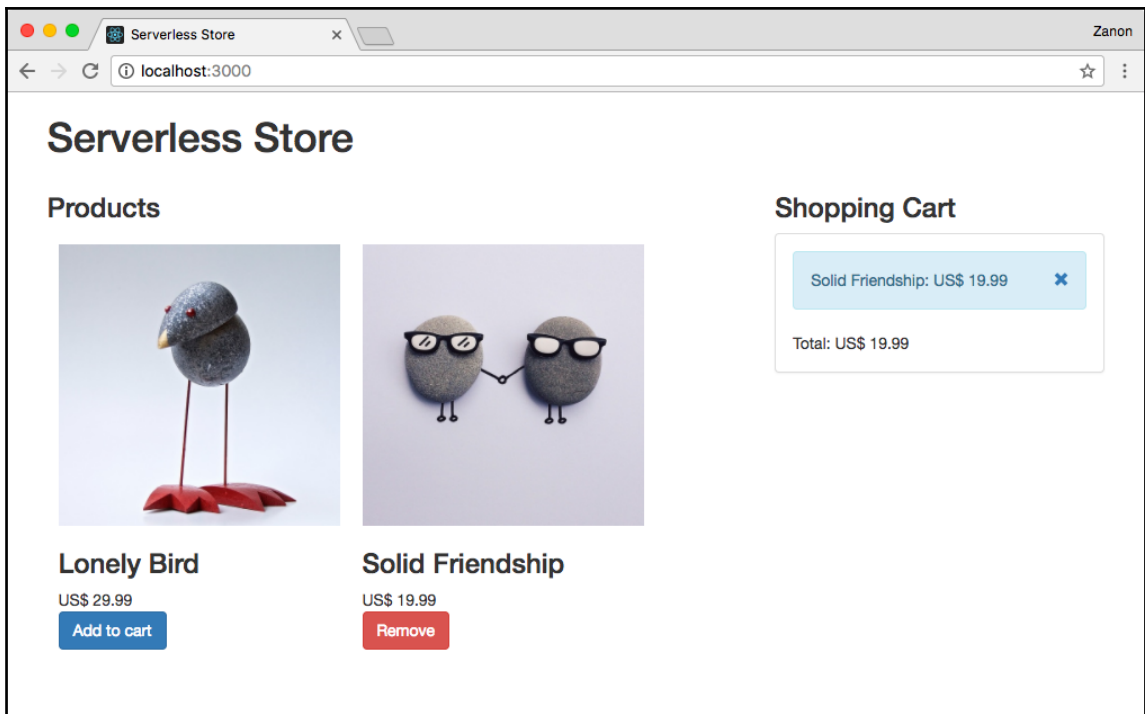
Edit Behavior

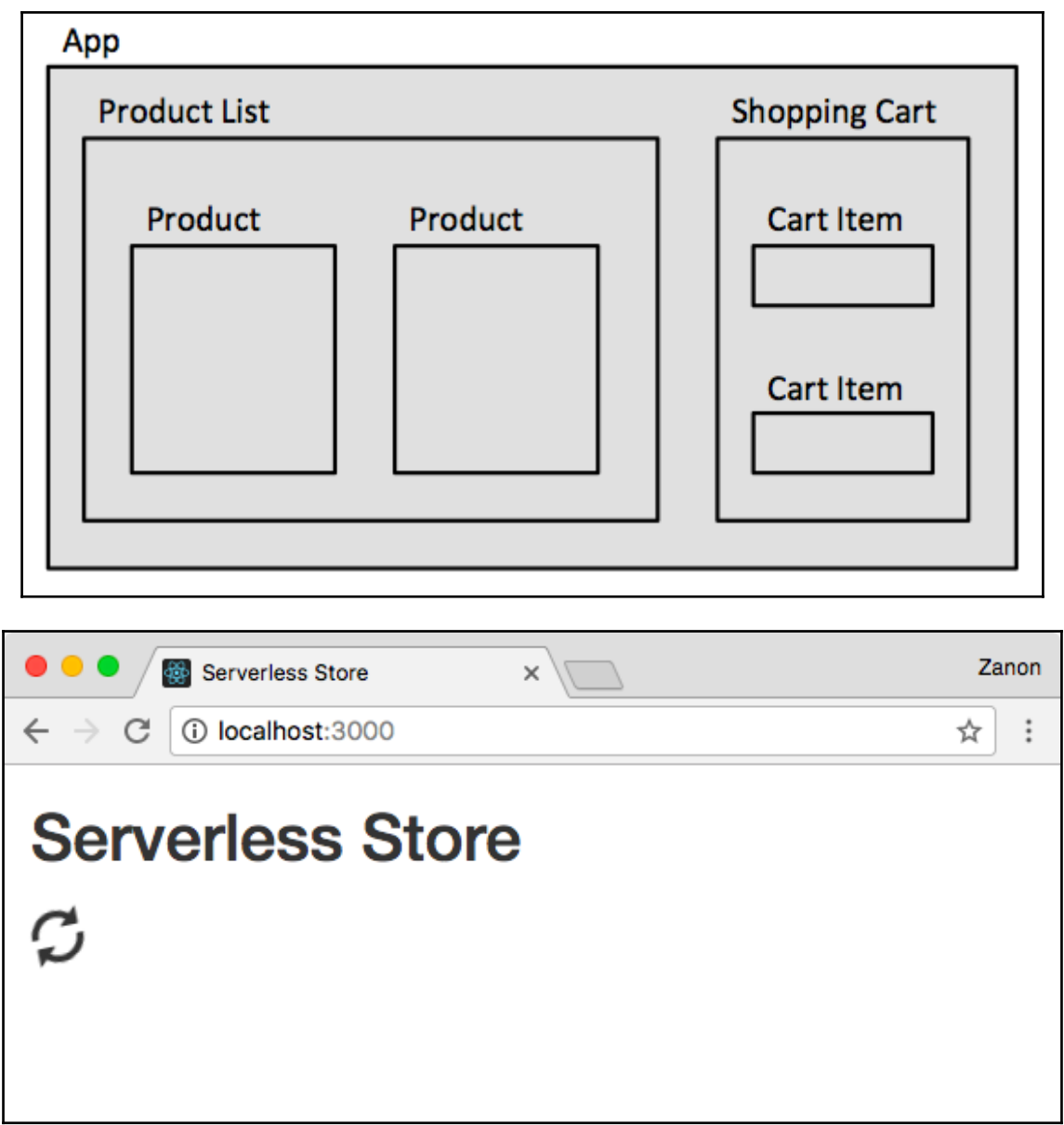
Default Cache Behavior Settings

Path Pattern	Default (*)	
Origin	S3-Website-serverless-store.zanon.io.s3-websit ▼	
Viewer Protocol Policy	<div><input type="radio"/> HTTP and HTTPS</div> <div><input checked="" type="radio"/> Redirect HTTP to HTTPS</div> <div><input type="radio"/> HTTPS Only</div>	
Allowed HTTP Methods	<div><input type="radio"/> GET, HEAD</div> <div><input type="radio"/> GET, HEAD, OPTIONS</div> <div><input checked="" type="radio"/> GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE</div>	
Cached HTTP Methods	GET, HEAD (Cached by default) <input checked="" type="checkbox"/> OPTIONS	
Forward Headers	None (Improves Caching) ▼	
Object Caching	<div><input checked="" type="radio"/> Use Origin Cache Headers</div> <div><input type="radio"/> Customize</div>	

[Learn More](#)

Chapter 5: Building the Frontend






Serverless Store

localhost:3000

Zanon

Serverless Store


Products



A sculpture of a single grey stone bird with a yellow beak and red eyes, standing on a red, multi-lobed base.

Lonely Bird

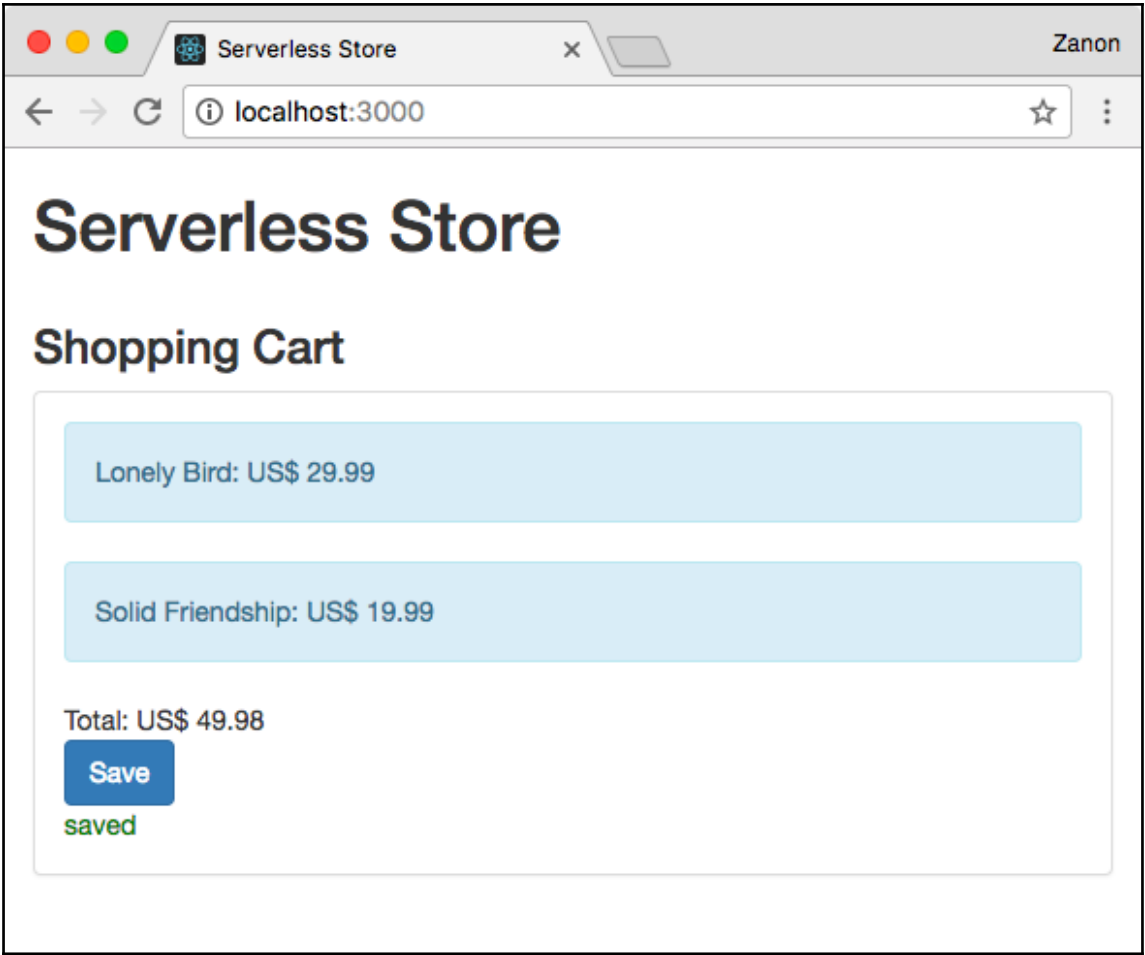
US\$ 29.99

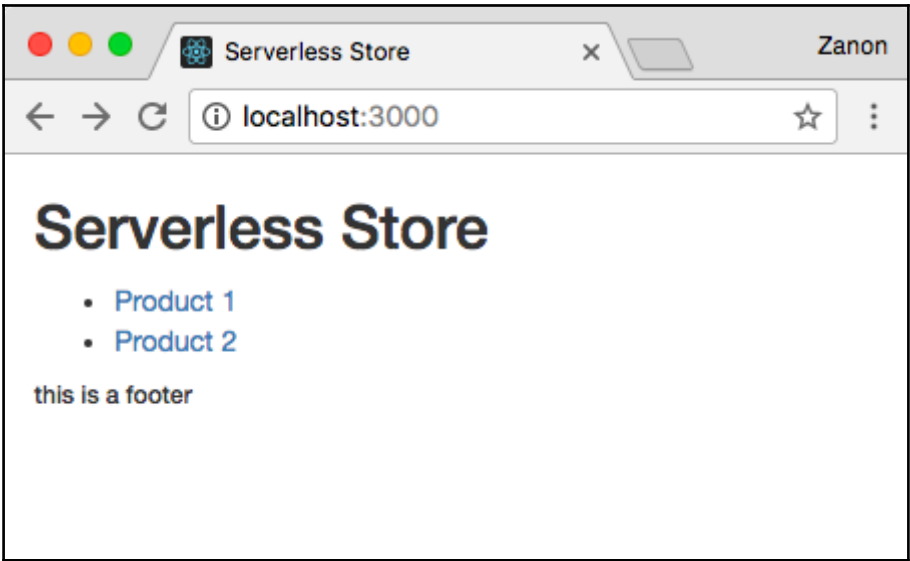
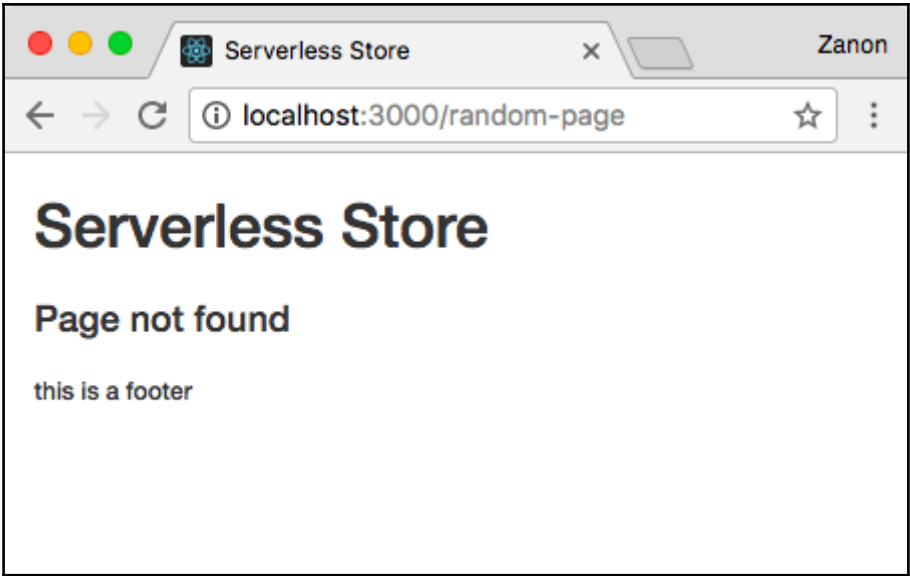


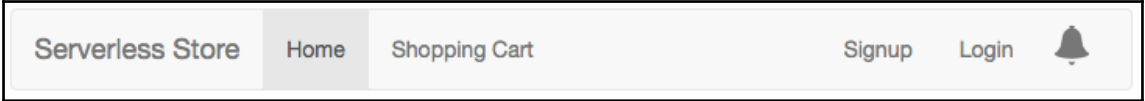
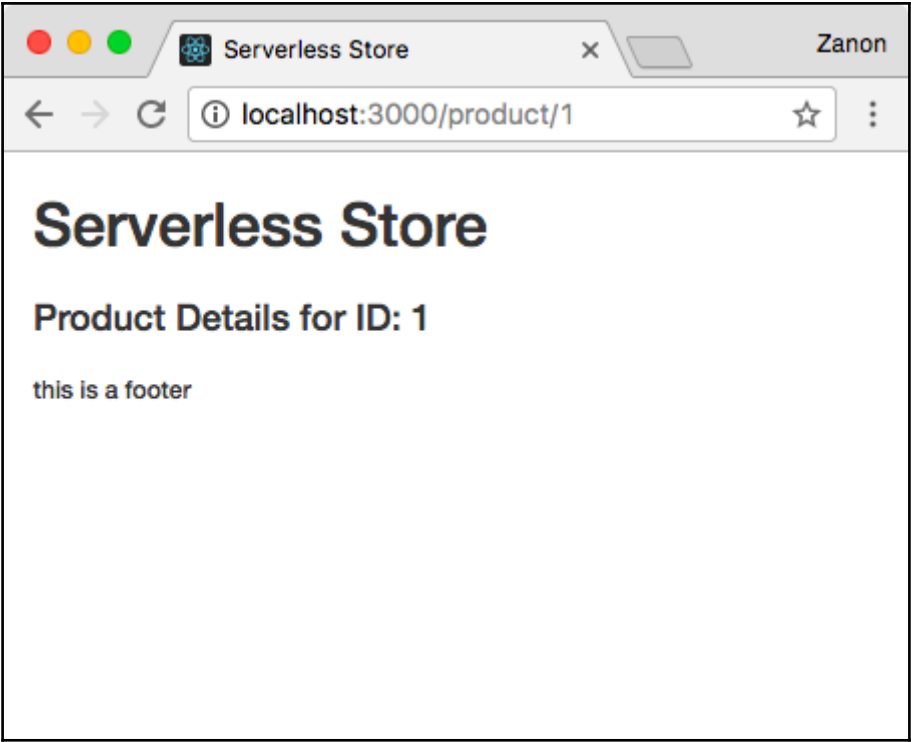
A sculpture of two grey stone figures wearing black-rimmed glasses, holding hands.

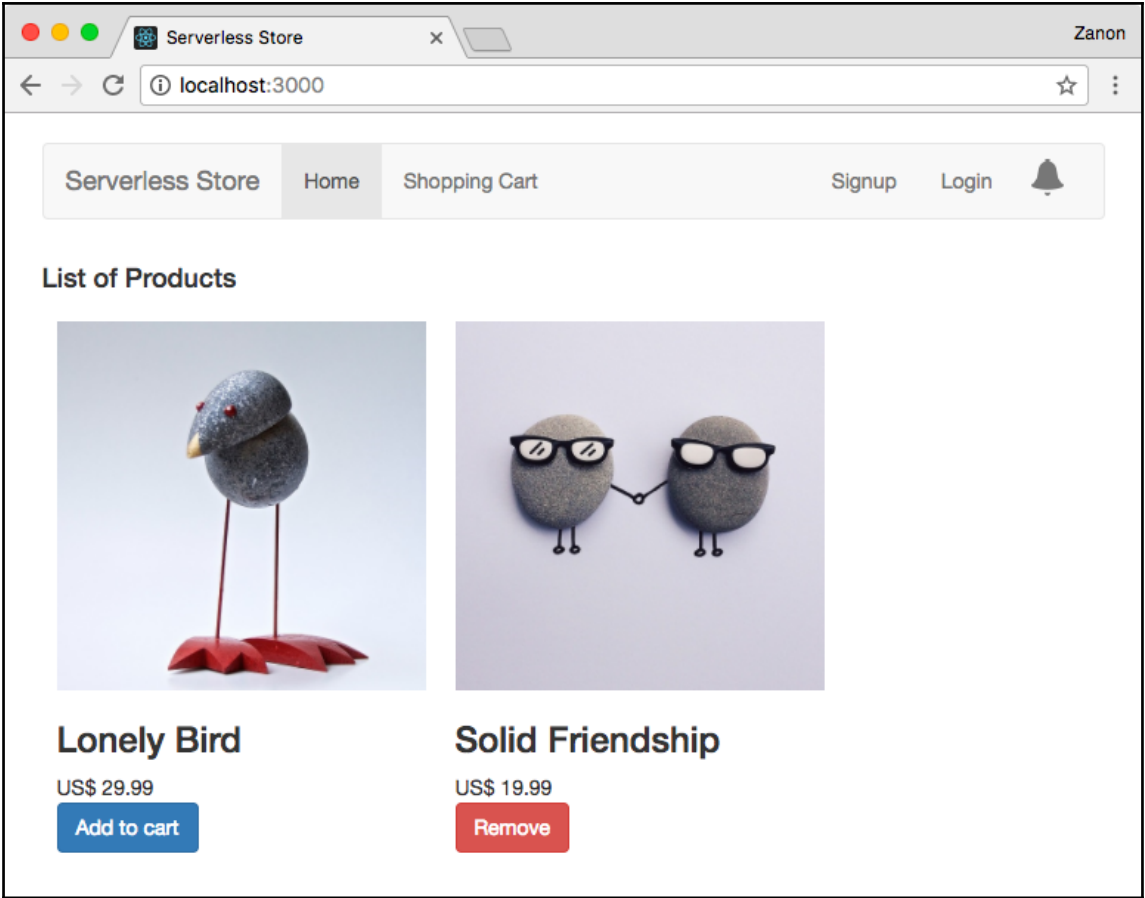
Solid Friendship

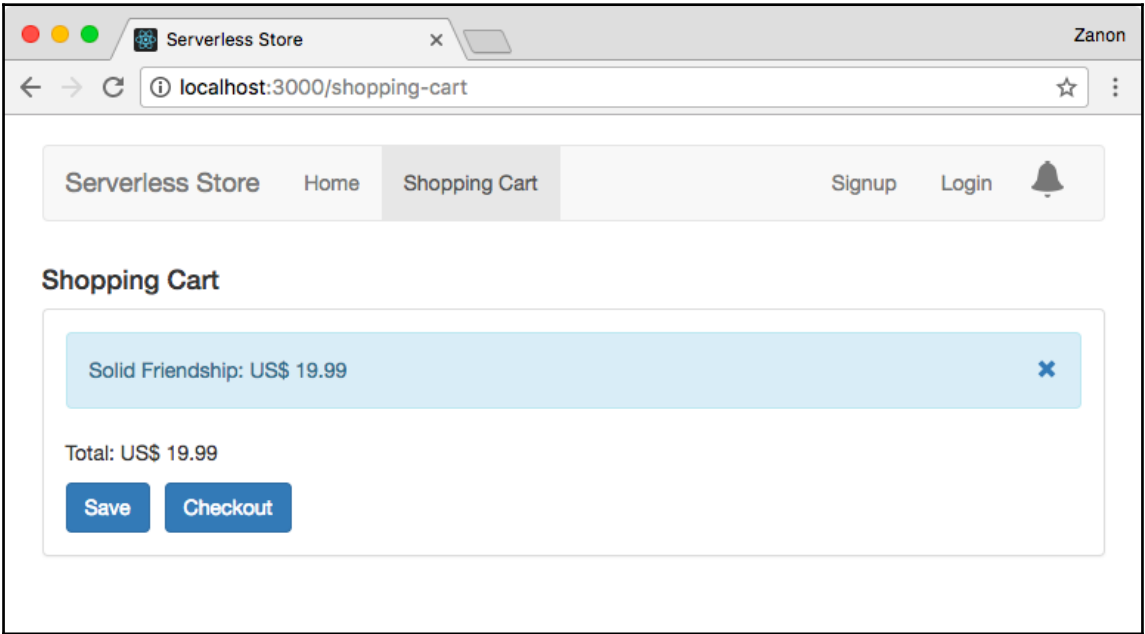
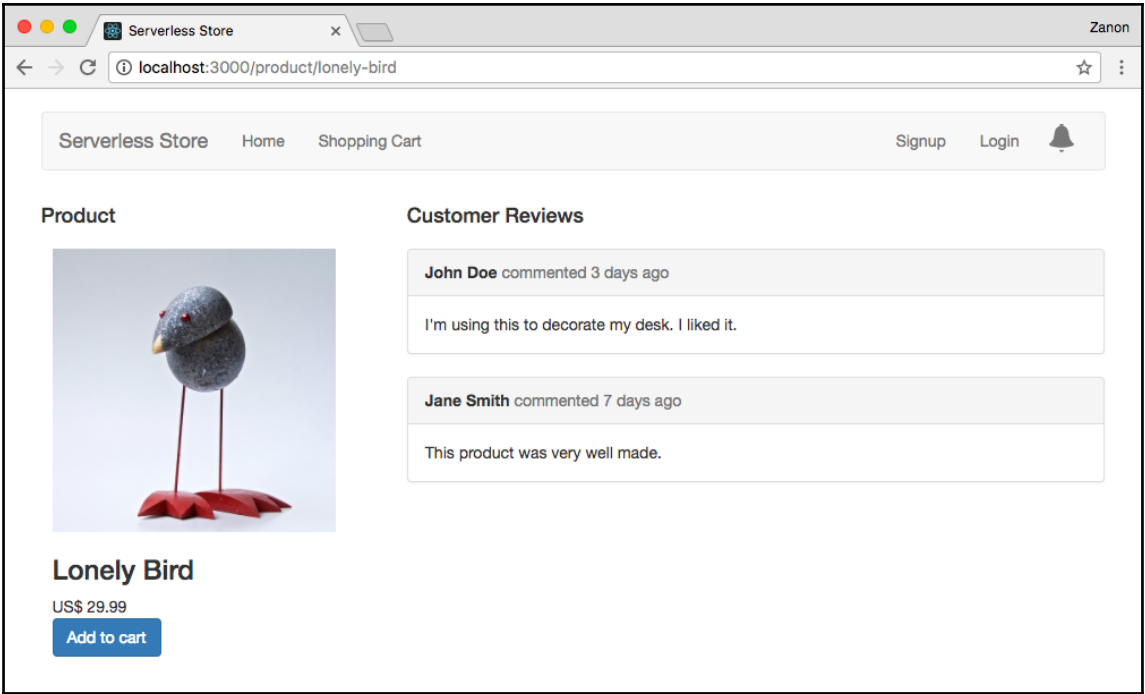
US\$ 19.99

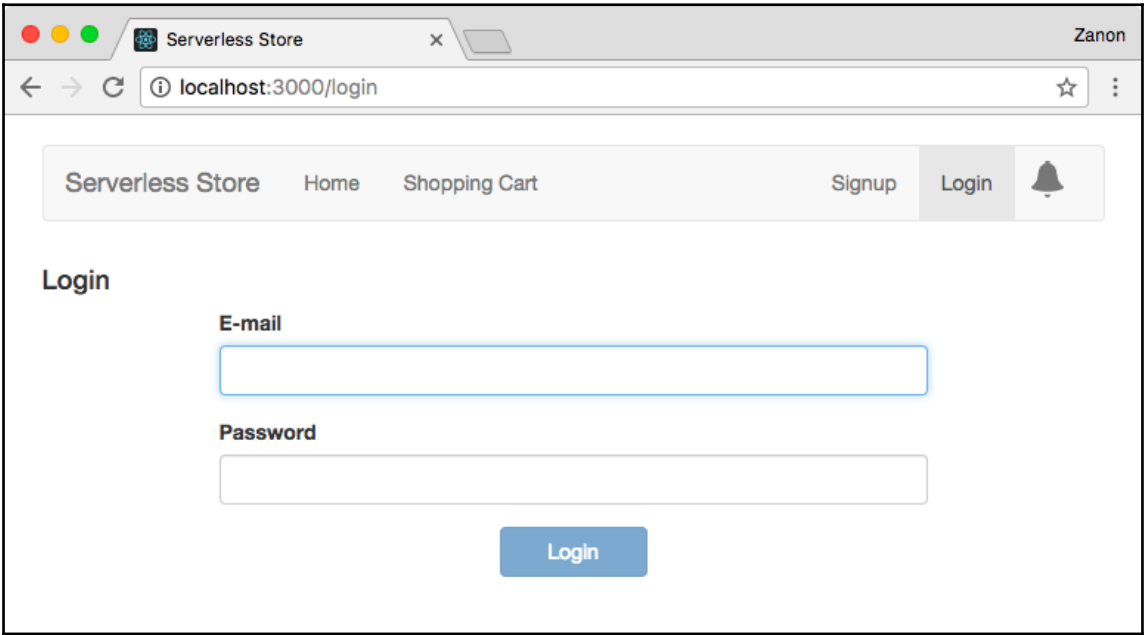
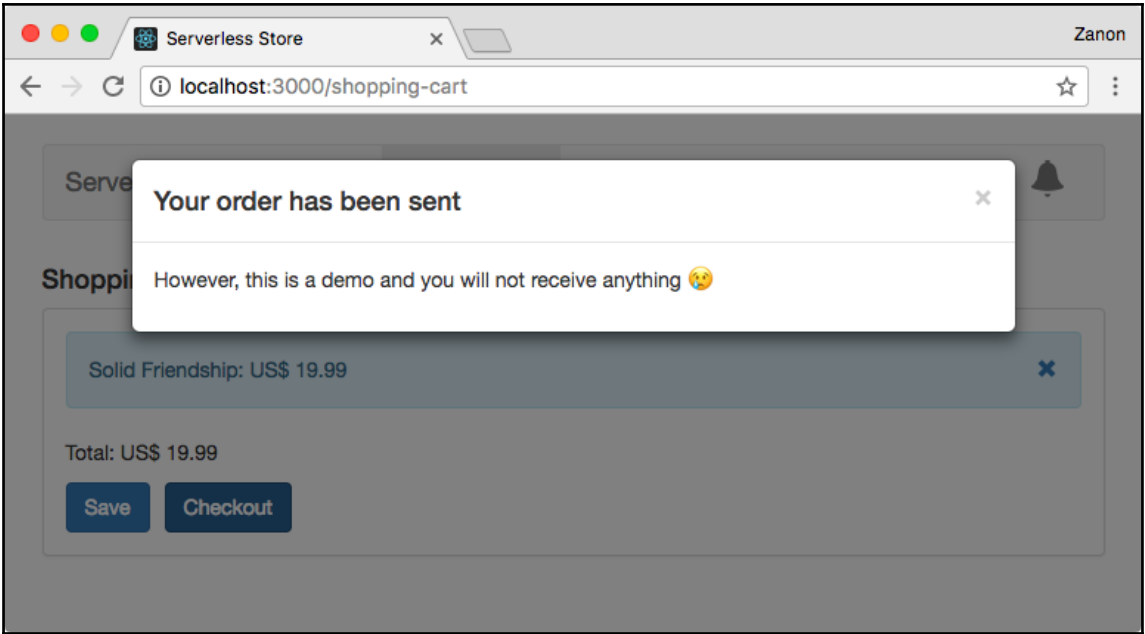


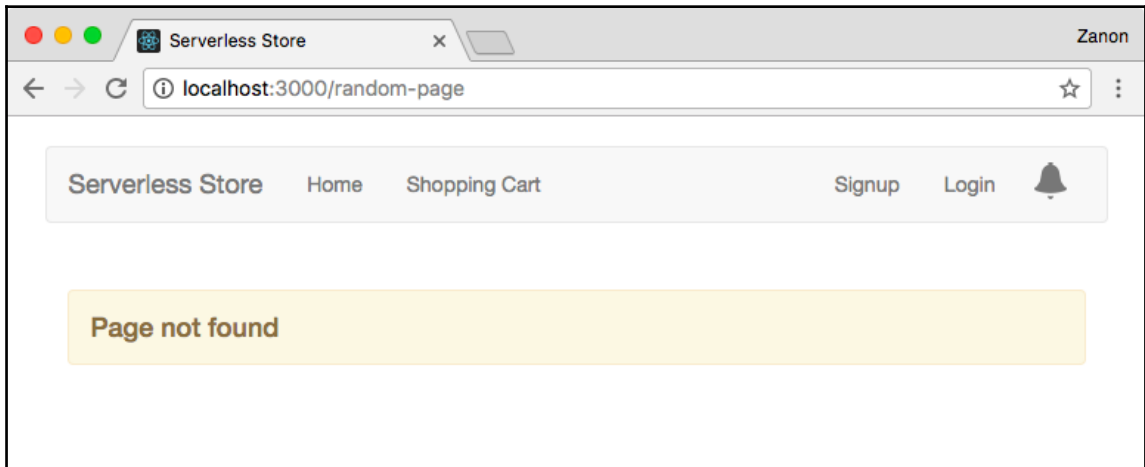
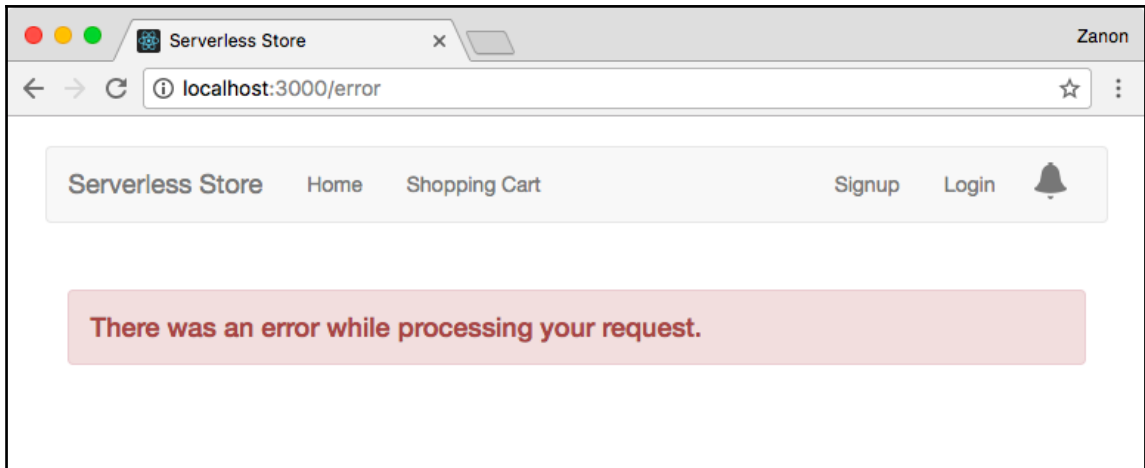




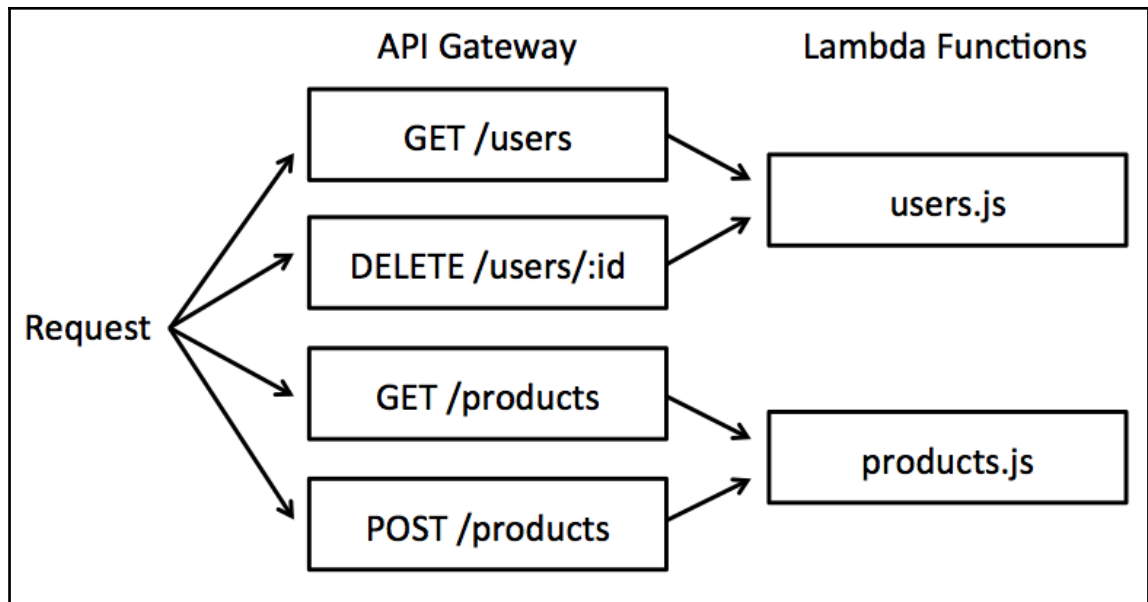
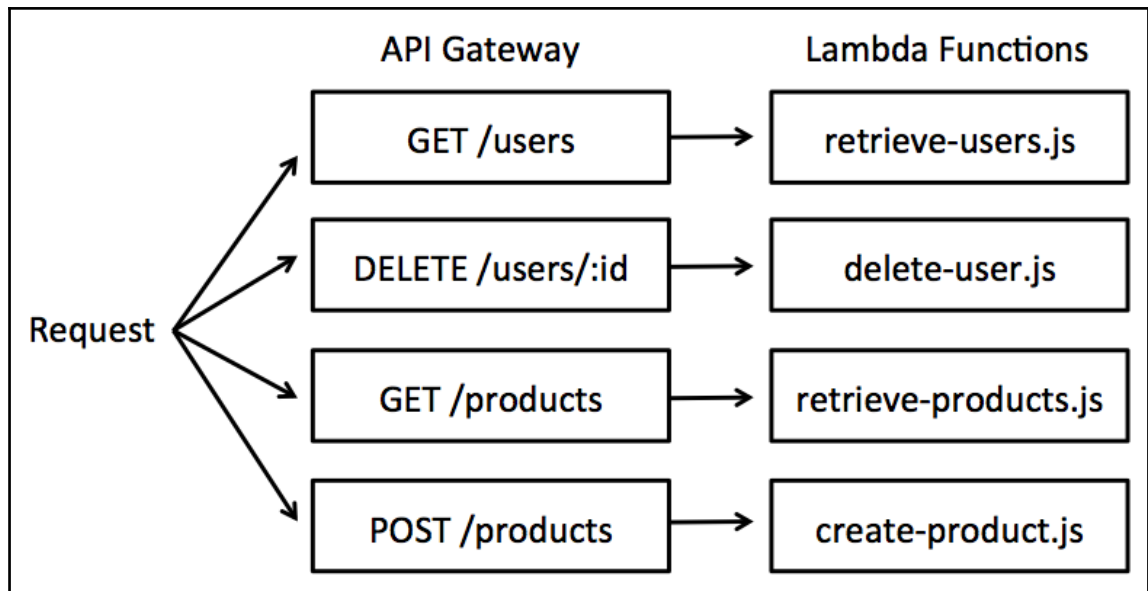


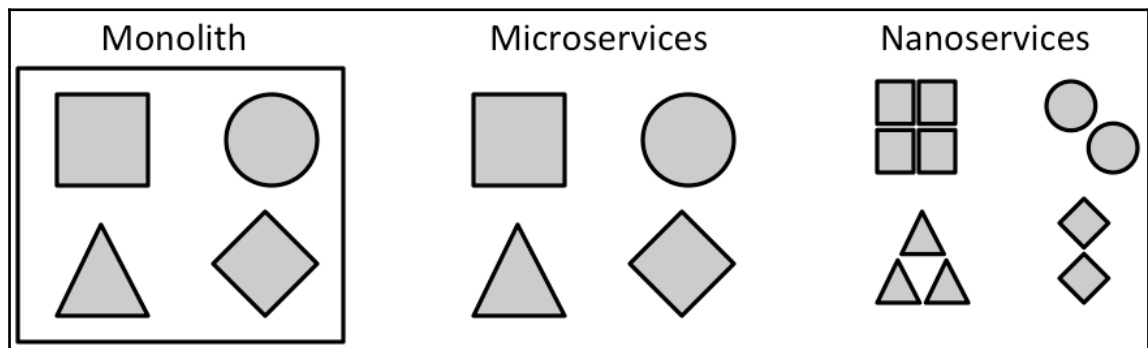
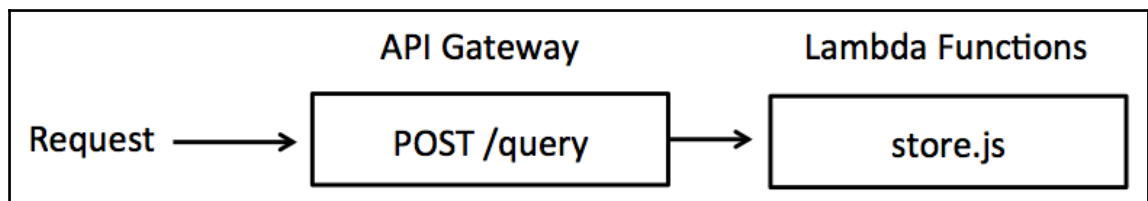
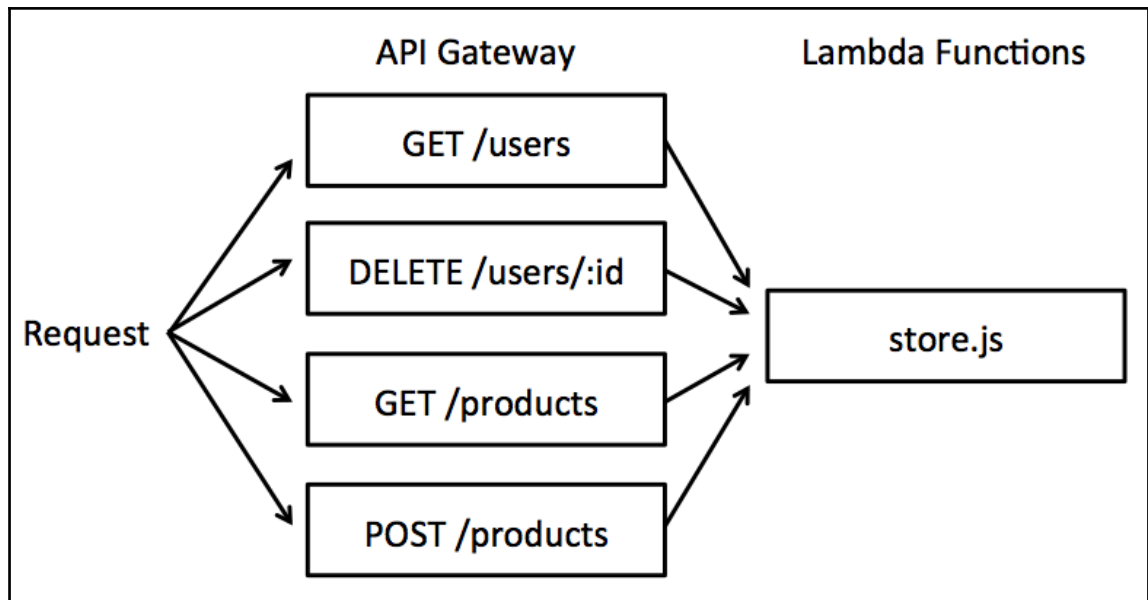






Chapter 6: Developing the Backend

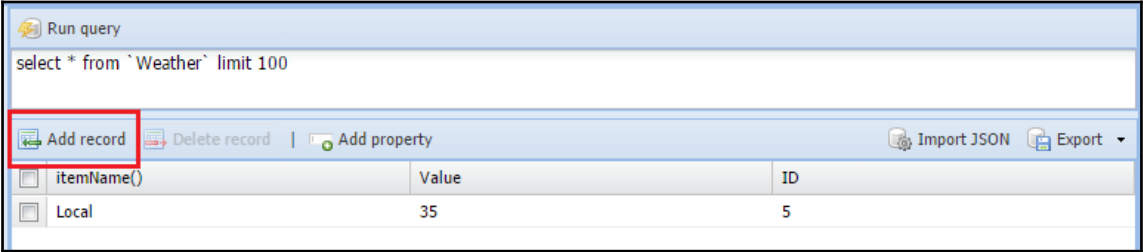
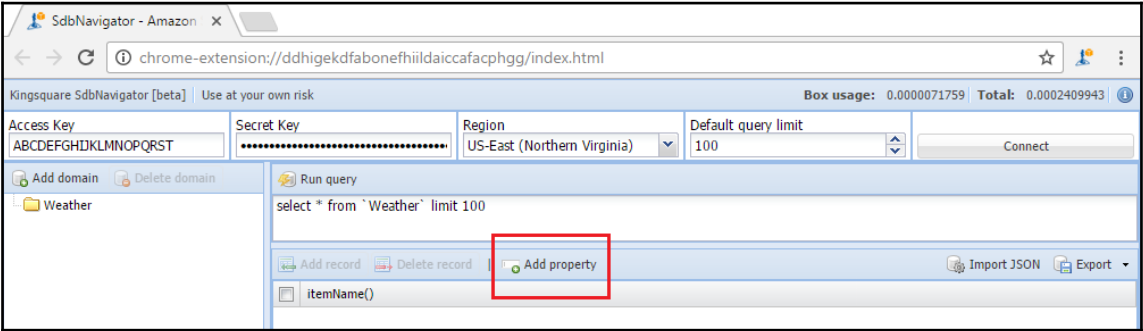
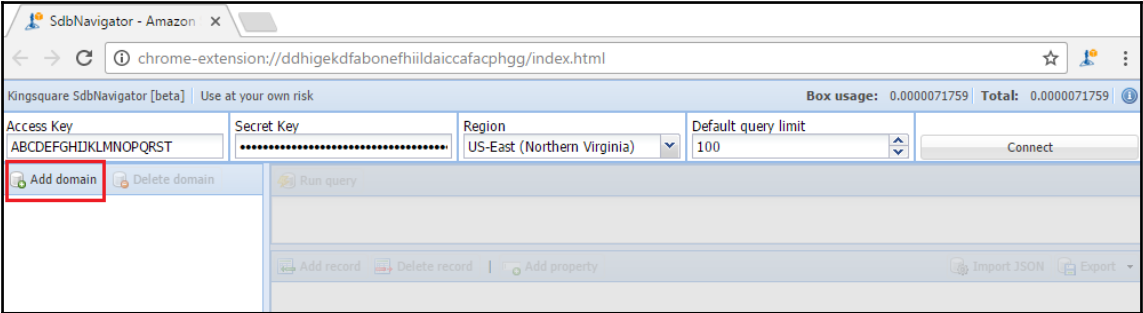




```
└─ BACKEND
  └─ functions
    └─ products.js
  └─ lib
    └─ cart.js
    └─ checkout.js
    └─ products.js
    └─ utils.js
  └─ node_modules
  └─ repositories
    └─ dynamodb.js
    └─ fakedb.js
    └─ simpledb.js
  └─ test
    └─ integration
    └─ unit
  └─ package.json
  └─ serverless.yml
```

```
└─ USING-MULTIPLE-SERVICES
  └─ service1
    └─ greetings.js
    └─ handler.js
    └─ package.json
    └─ serverless.yml
  └─ service2
    └─ handler.js
    └─ package.json
    └─ serverless.yml
```

Chapter 7: Managing a Serverless Database



Services

Resource Groups

DynamoDB

Dashboard

Tables

Reserved capacity

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

Create table

Amazon DynamoDB is a fully managed non-relational database service that provides fast and predictable performance with seamless scalability.

Create table

Recent alerts

No CloudWatch alarms have been triggered.

View all in CloudWatch

Total capacity for US East (N. Virginia)

Provisioned read capacity10

Reserved read capacity0

Provisioned write capacity10

Reserved write capacity0

Service health

Current Status

Details

Amazon DynamoDB (N. Virginia)

Service is operating normally

View complete service health details

Services

Resource Groups

Diego Zanon

N. Virginia

Support

DynamoDB

Dashboard

Tables

Reserved capacity

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

Create table

Actions

Filter by table name

Name

Products

ShoppingCart

Products

Close

Overview

Items

Metrics

Alarms

Capacity

Indexes

More

Create item

Actions

Scan: [Table] Products: ID

Viewing 1 to 1 items

Scan

[Table] Products: ID

Add filter

Start search

ID

Comments

Image

Name

Price

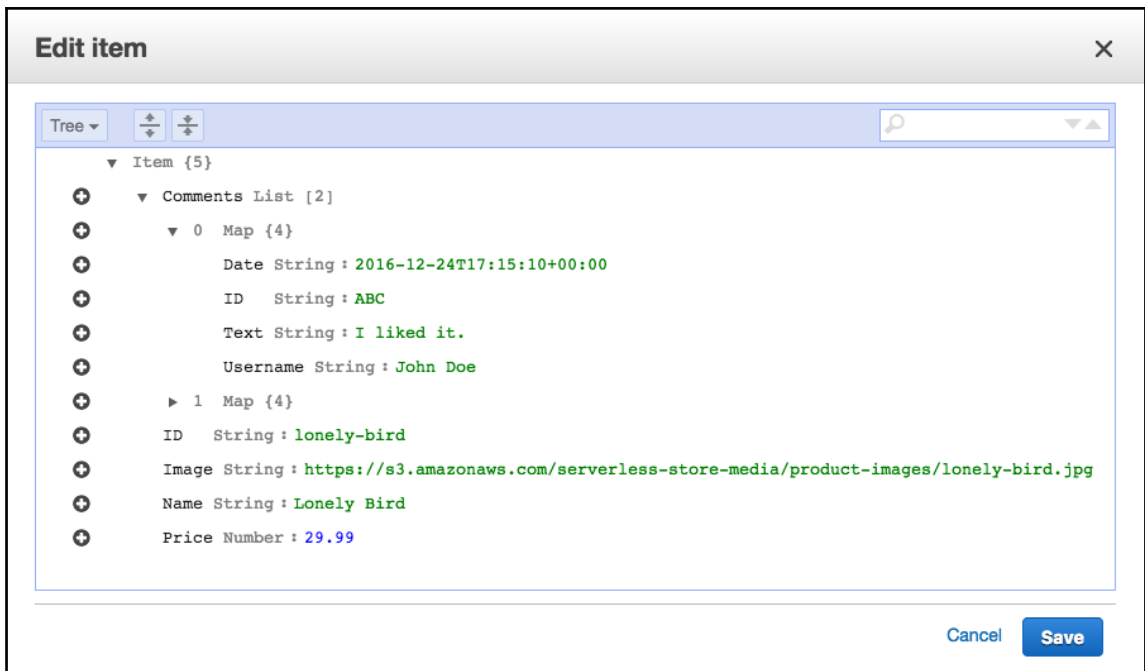
lonely-bird

[{"M": {"Da...

https://s3.am...

Lonely Bird

29.99



Auto Scaling

☒ Read capacity

Target utilization

70

%

Minimum provisioned capacity

5

units

Maximum provisioned capacity

40000

units

☐ Apply same settings to global secondary indexes

☒ Write capacity

☐ Same settings as read

70

%

5

units

40000

units

☐ Apply same settings to global secondary indexes

IAM Role

I authorize DynamoDB to scale capacity using the following role:

☒ New role: DynamoDBAutoscaleRole

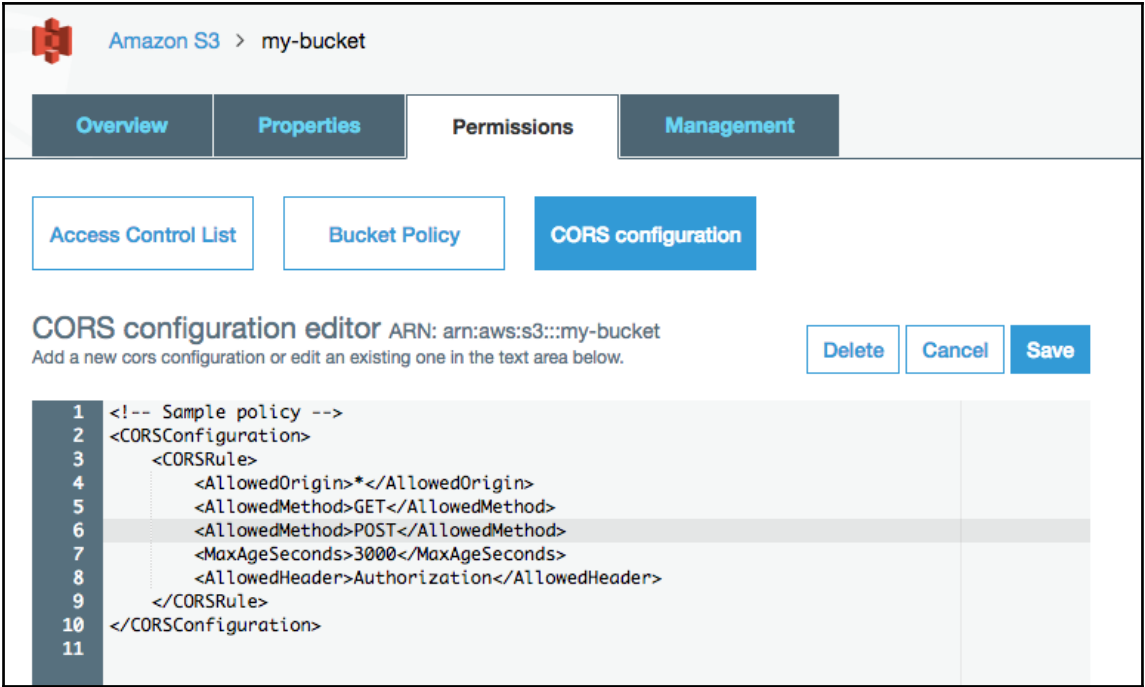
☐ Existing role with pre-defined policies [\[Instructions\]](#)

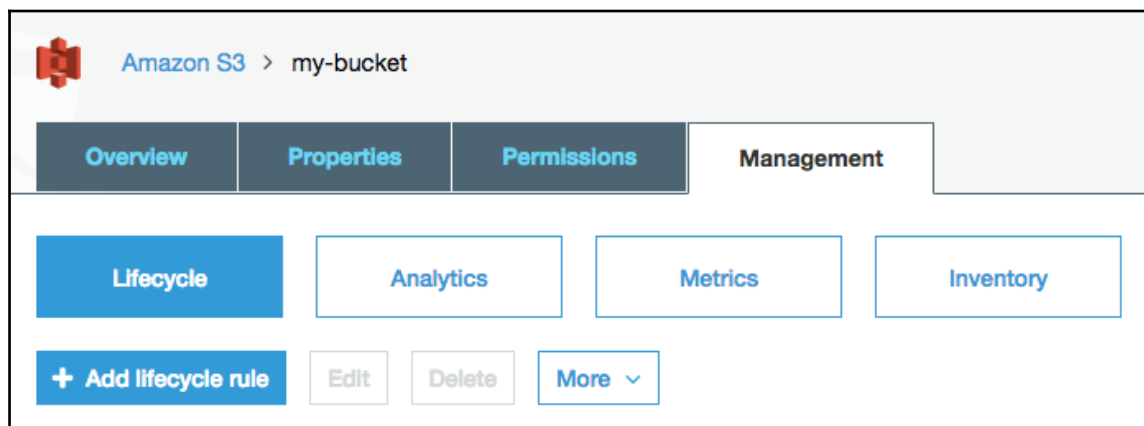
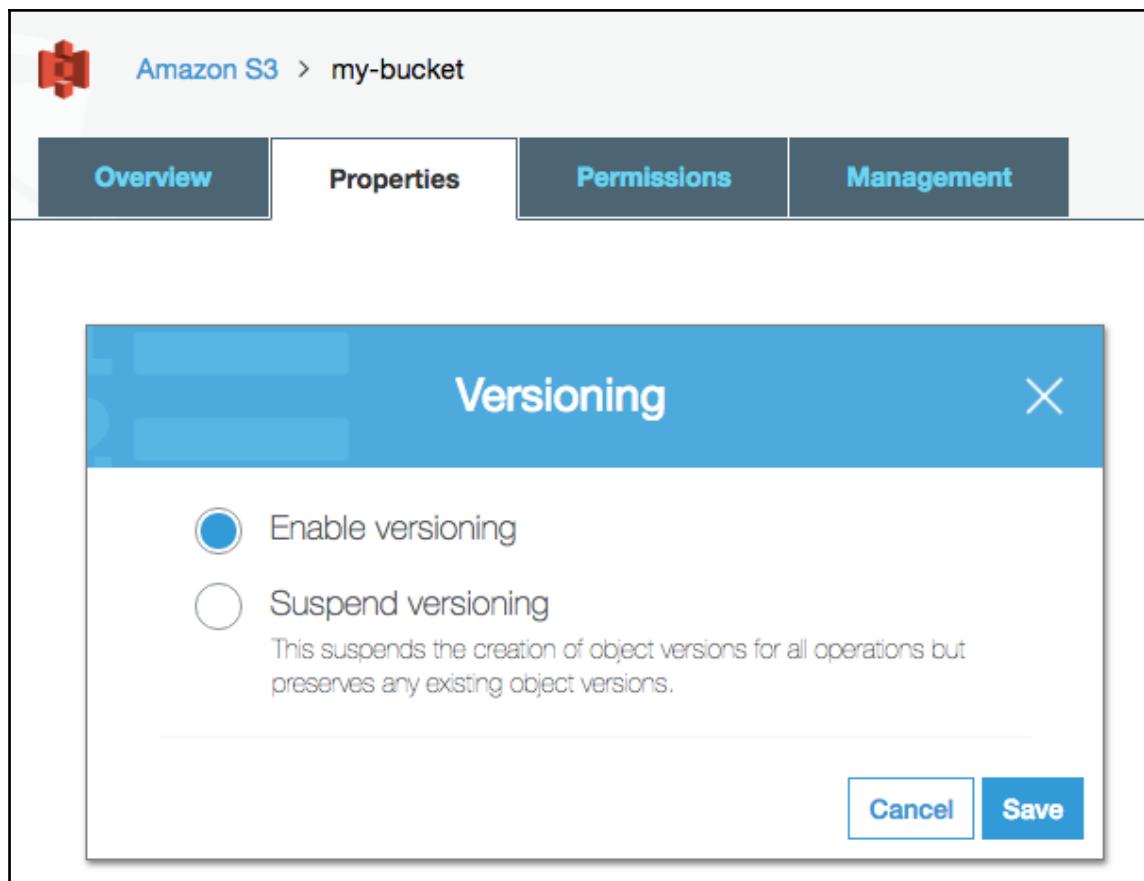
Role Name*

Save

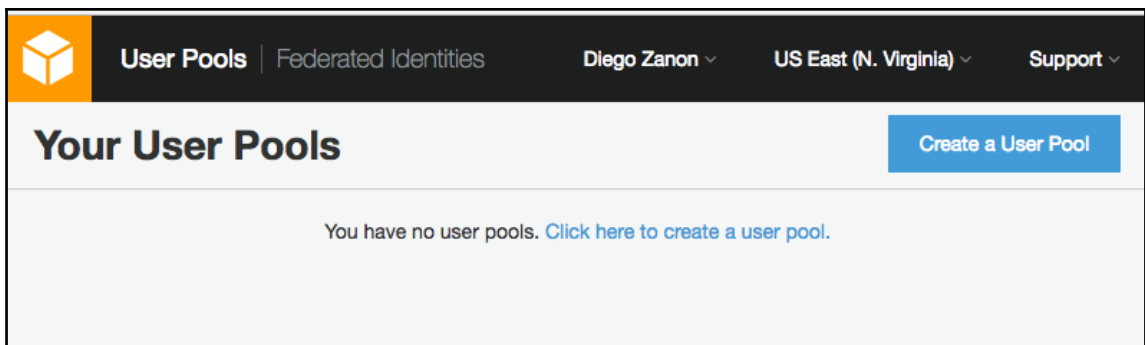
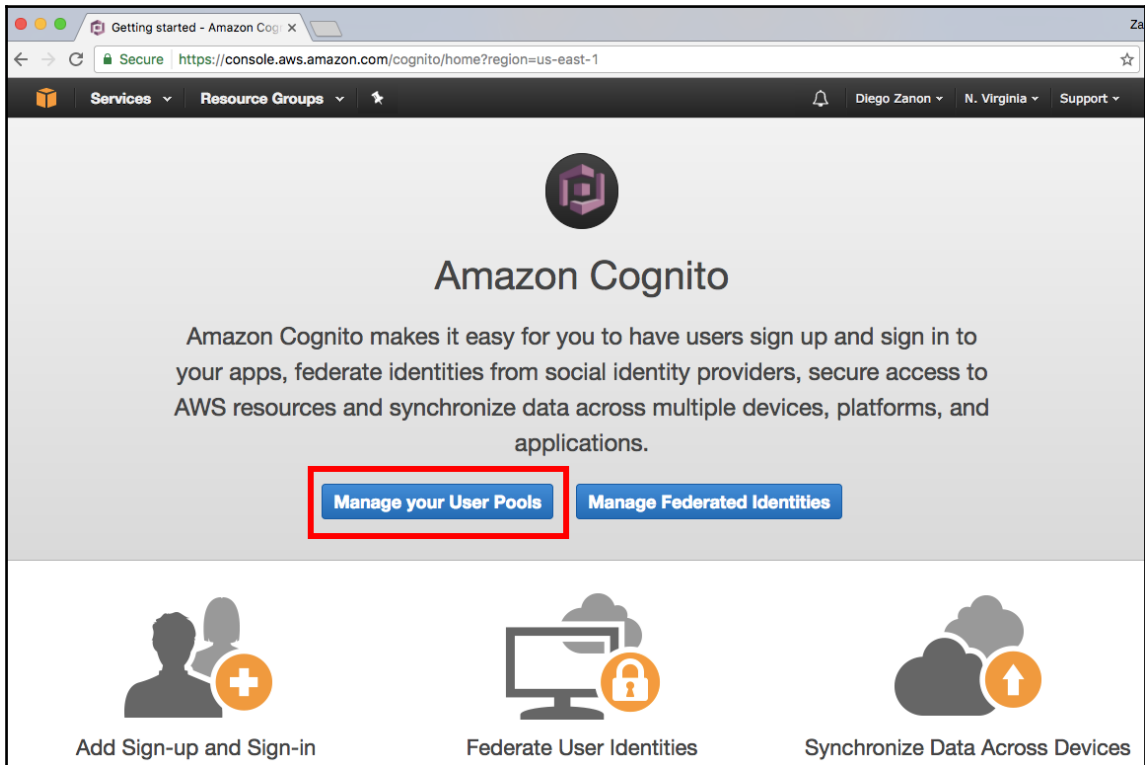
Cancel

[66]





Chapter 8: Securing the Serverless Application



User Pools

Federated Identities

Diego Zanon

US East (N. Virginia)

Support

Create a user pool

Cancel

Name

Attributes

Policies

Verifications

Message customizations

Tags

Devices

Apps

Triggers

Review

What do you want to name your user pool?

Give your user pool a descriptive name so you can easily identify it in the future.

Pool name

serverless-store

How do you want to create your user pool?

Review defaults

Start by reviewing the defaults and then customize as desired

Step through settings

Step through each setting to make your choices

[70]

Pool Nameserverless-store

Required attributesemail

Alias attributesChoose alias attributes...

Custom attributesChoose custom attributes...

Minimum password length8

Password policyuppercase letters, lowercase letters, special characters, numbers

User sign ups allowed?Users can sign themselves up

MFAEnable MFA...

Verificationsemail

TagsChoose Tags for Your User Pool...

AppsAdd client...

TriggersAdd triggers...

Create pool

Your user pool was created successfully.

Pool Idus-east-1_aBcdeFghi

Pool ARNarn:aws:cognito-idp:us-east-1:1234567890:userpool/us-east-1_aBcdeFghi

Estimated number of users0

Which app clients will have access to this user pool?

The app clients that you add below will be given a unique ID and an optional secret key to access this user pool.

App client name

Refresh token expiration (days)

☐ Generate client secret

☐ Enable sign-in API for server-based authentication (ADMIN_NO_SRP_AUTH) [Learn more.](#)

☐ Only allow Custom Authentication (CUSTOM_AUTH_FLOW_ONLY) [Learn more.](#)

[Set attribute read and write permissions](#)

Which app clients will have access to this user pool?

The app clients that you add below will be given a unique ID and an optional secret key to access this user pool.

×

App client name

serverless-store

App client id

abcdefghijklmnopqrstuvwxyz

Show Details

The image shows a screenshot of the AWS IAM console. At the top, there is a navigation bar with the AWS logo and two tabs: 'User Pools' and 'Federated Identities'. The 'Federated Identities' tab is selected and highlighted with a red rectangular box. Below the navigation bar, the main content area displays the name of the user pool, 'serverless-store', in a large, bold font. At the bottom of the screenshot, there is a section titled 'Pool details' with a light orange background.

[73]

The screenshot shows the AWS Cognito console interface for creating a new identity pool. The browser address bar shows the URL `https://console.aws.amazon.com/cognito/create`. The page title is "Getting started wizard". On the left, a sidebar indicates "Step 1: Create identity pool" and "Step 2: Set permissions". The main content area is titled "Create new identity pool" and includes instructions: "Identity pools are used to store end user identities. To declare a new identity pool, enter a unique name." Below this, the "Identity pool name*" field contains the text "serverless_store" and is marked as valid with a green checkmark. An example "Example: My App Name" is shown. A section titled "Unauthenticated identities" explains that Amazon Cognito can support unauthenticated identities. A checkbox labeled "Enable access to unauthenticated identities" is checked and highlighted with a red rectangular box. Below this is a section for "Authentication providers". At the bottom, there is a "* Required" label and two buttons: "Cancel" and "Create Pool". The footer contains links for "Feedback", "English", "Privacy Policy", and "Terms of Use", along with copyright information for Amazon Web Services.

New identity pool x Zanon

Secure `https://console.aws.amazon.com/cognito/create`

Services Resource Groups

Diego Zanon Select a Region Support

Getting started wizard

Step 1: Create identity pool
Step 2: Set permissions

Create new identity pool

Identity pools are used to store end user identities. To declare a new identity pool, enter a unique name.

Identity pool name* ✓
Example: My App Name

▼ Unauthenticated identities ⓘ

Amazon Cognito can support unauthenticated identities by providing a unique identifier and AWS credentials for users who do not authenticate with an identity provider. If your application allows customers to use the application without logging in, you can enable access for unauthenticated identities. [Learn more about unauthenticated identities.](#)

☒ Enable access to unauthenticated identities

► Authentication providers ⓘ

* Required Cancel Create Pool

Feedback English © 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

▼ Authentication providers ⓘ

Amazon Cognito supports the following authentication methods with Amazon Cognito Sign-In or any public provider. If you allow your users to authenticate using any of these public providers, you can specify your application identifiers here. Warning: Changing the application ID that your identity pool is linked to will prevent existing users from authenticating using Amazon Cognito. [Learn more about public identity providers.](#)

Cognito

Amazon

Facebook

Google+

Twitter / Digits

OpenID

SAML

Custom

Configure your Cognito Identity Pool to accept users federated with your Cognito User Pool by supplying the User Pool ID and the App Client ID.

User Pool ID

Optional

ex: us-east-1_Ab129faBb

App Client ID

Optional

ex: 7lhkkfbfb4q5kpp90urffao

Add Another Provider

▼ Hide Details

Role Summary ?

Role

Your authenticated identities would like access to Cognito.

Description

IAM Role

Create a new IAM Role

⬆⬇⬆

Role Name

Cognito_serverless_storeAuth_Rol

▶ View Policy Document

Role Summary ?

Role

Your unauthenticated identities would like access to Cognito.

Description

IAM Role

Create a new IAM Role

⬆⬇⬆

Role Name

Cognito_serverless_storeUnauth_f

▶ View Policy Document

[76]

The screenshot shows the AWS IAM console interface. The top navigation bar includes 'Services', 'Resource Groups', and user information. The left sidebar shows 'Federated Identities' with a sub-menu for 'serverless_store'. The main content area is titled 'Edit identity pool'. It contains a description of identity pools and their roles. Below the description, there are three fields: 'Identity pool name*' (serverless_store), 'Identity pool ID' (us-east-1:abcdefghi-1234-5678-9012-abcdefghijkl), and 'Unauthenticated role' (Cognito_serverless_storeUnauth_Role). The 'Identity pool ID' field is highlighted with a red rectangle. Below the 'Unauthenticated role' field, there is an 'Authenticated role' field (Cognito_serverless_storeAuth_Role).

Identity pool

- Dashboard
- Sample code
- Identity browser

Edit identity pool

From this page you can modify the details of your identity pool. An identity pool must have a unique name and a set of authenticated and unauthenticated roles. The roles are saved with your identity pool and whenever we receive a request to authorize a user we will automatically utilize the roles you specify here. You will be required to specify the identity pool id from this page when initializing the Amazon Cognito client SDK. [Learn more about using IAM roles with Amazon Cognito.](#)

Identity pool name*

Identity pool ID ⓘ (Show ARN)

Unauthenticated role ⓘ [Create new role](#)

Authenticated role ⓘ [Create new role](#)

The screenshot shows a web browser window with the title 'Serverless Store'. The address bar shows 'localhost:3000/signup'. The page has a navigation bar with 'Serverless Store', 'Home', 'Shopping Cart', 'Signup', 'Login', and a bell icon. The main content area is titled 'Signup' and contains three input fields: 'E-mail', 'Password', and 'Confirm Password'. A blue 'Signup' button is at the bottom.

Serverless Store Home Shopping Cart Signup Login

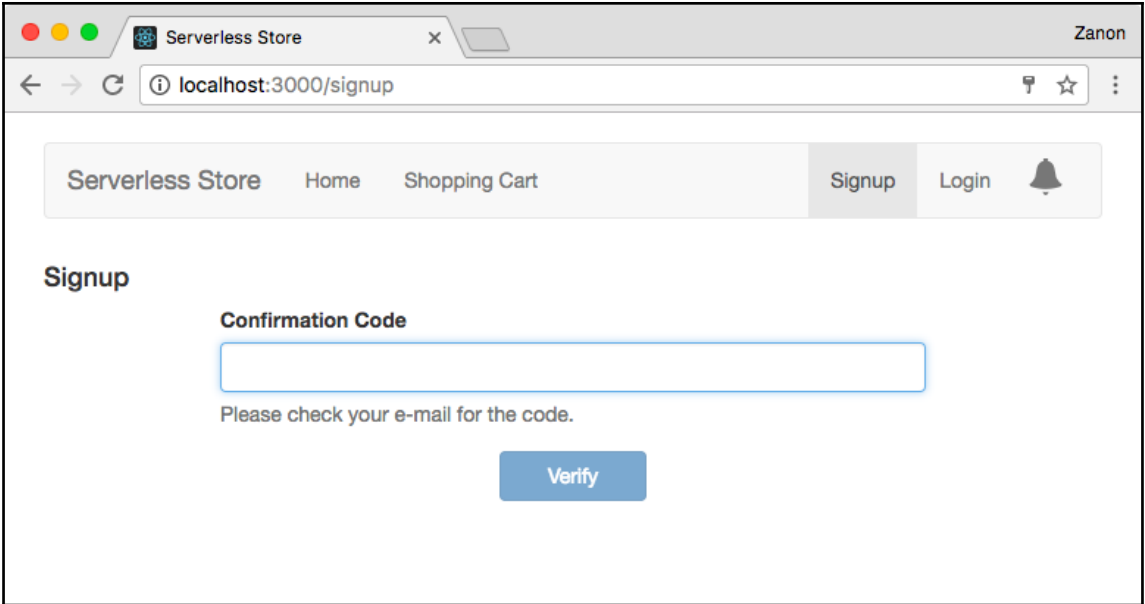
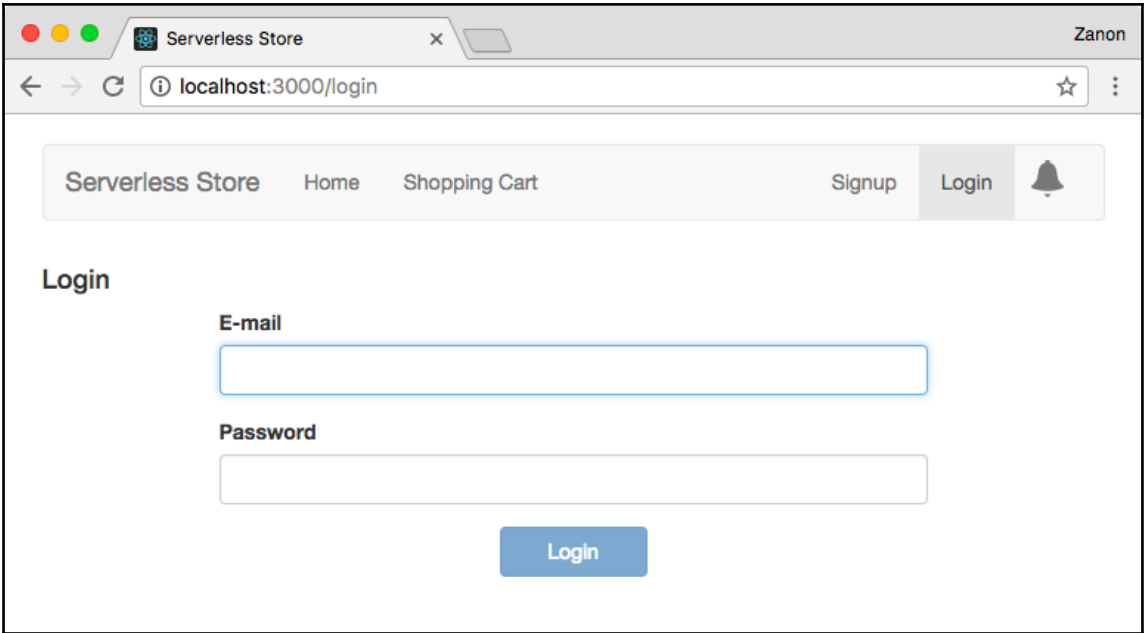
Signup

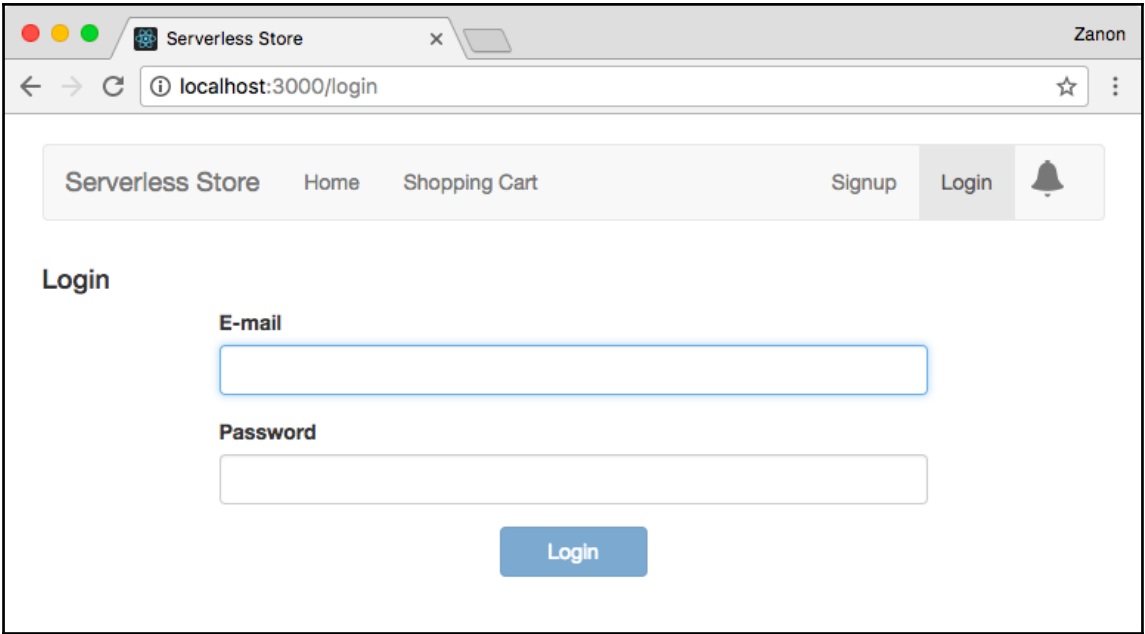
E-mail

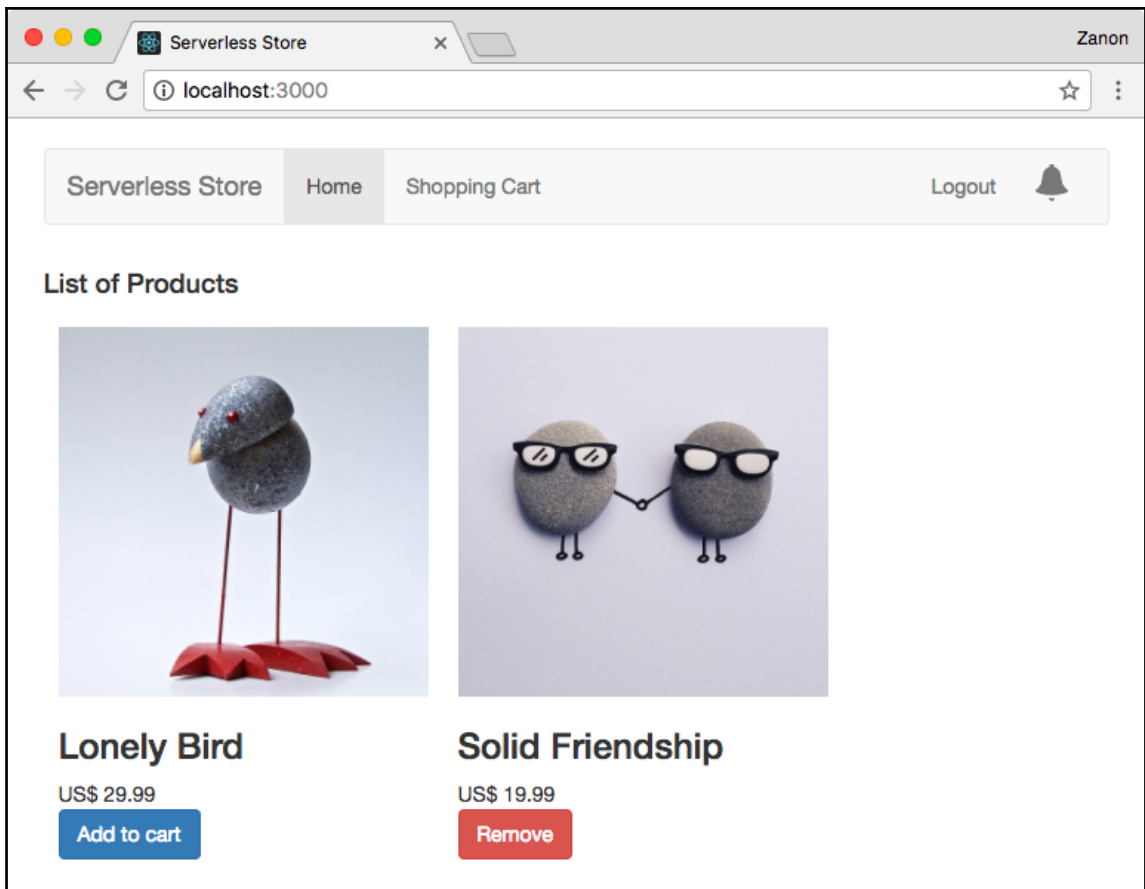
Password

Confirm Password

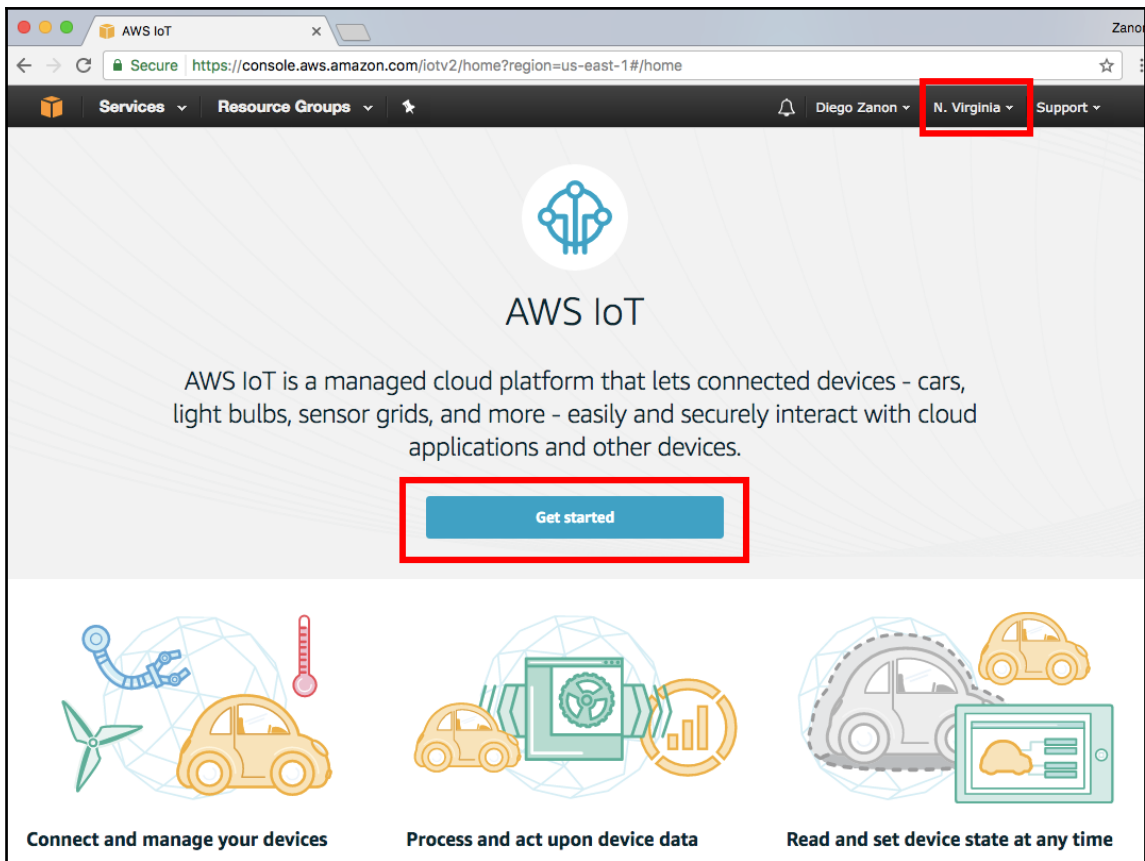
Signup

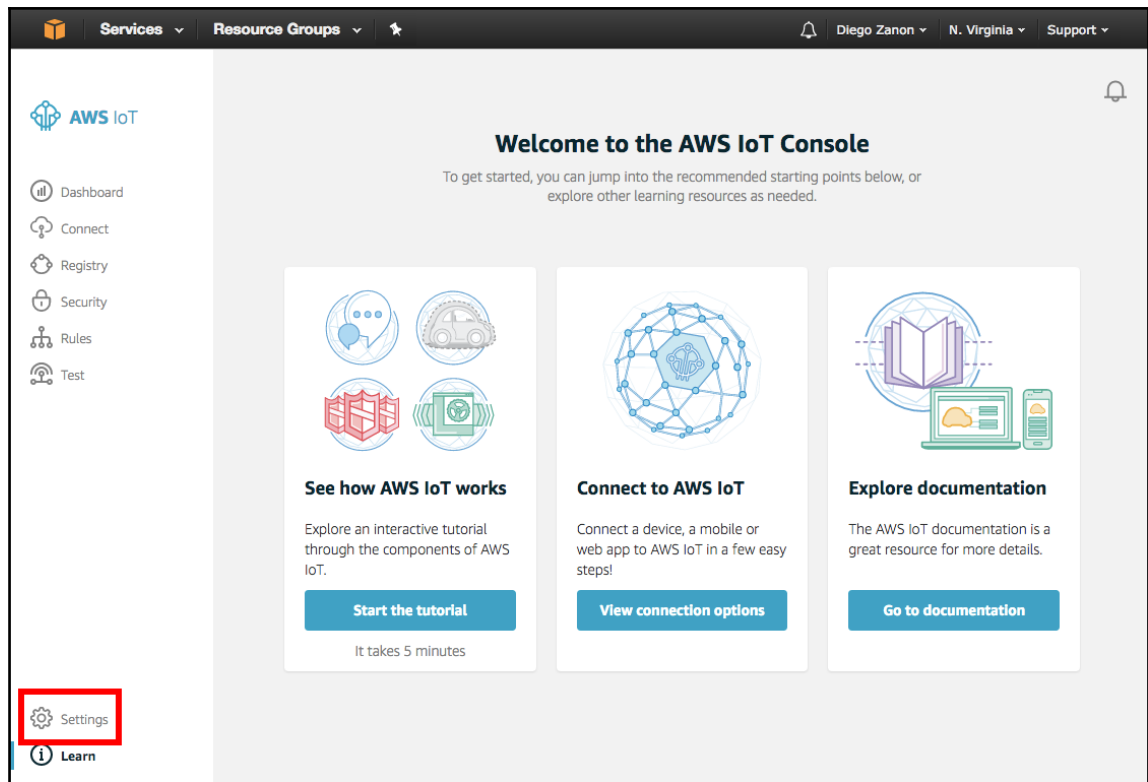






Chapter 9: Handling Serverless Notifications





Services ▾ **Resource Groups** ▾ ⭐


🔔 Diego Zanon ▾ N. Virginia ▾ Support ▾

AWS IoT

Dashboard
Connect
Registry
Security
Rules
Test
Settings
Learn

Welcome to the AWS IoT Console

To get started, you can jump into the recommended starting points below, or explore other learning resources as needed.




See how AWS IoT works

Explore an interactive tutorial through the components of AWS IoT.

[Start the tutorial](#)


It takes 5 minutes



Connect to AWS IoT

Connect a device, a mobile or web app to AWS IoT in a few easy steps!

[View connection options](#)



Explore documentation

The AWS IoT documentation is a great resource for more details.

[Go to documentation](#)

Custom endpoint

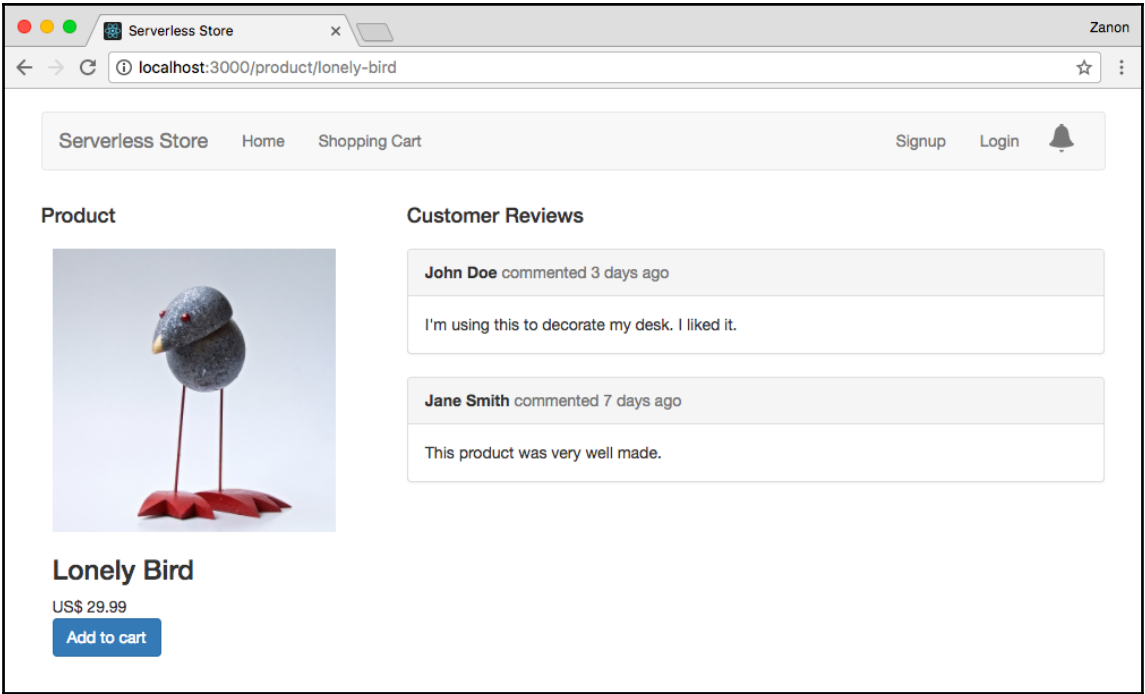
ENABLED

This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT [Device SDK](#).

Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.

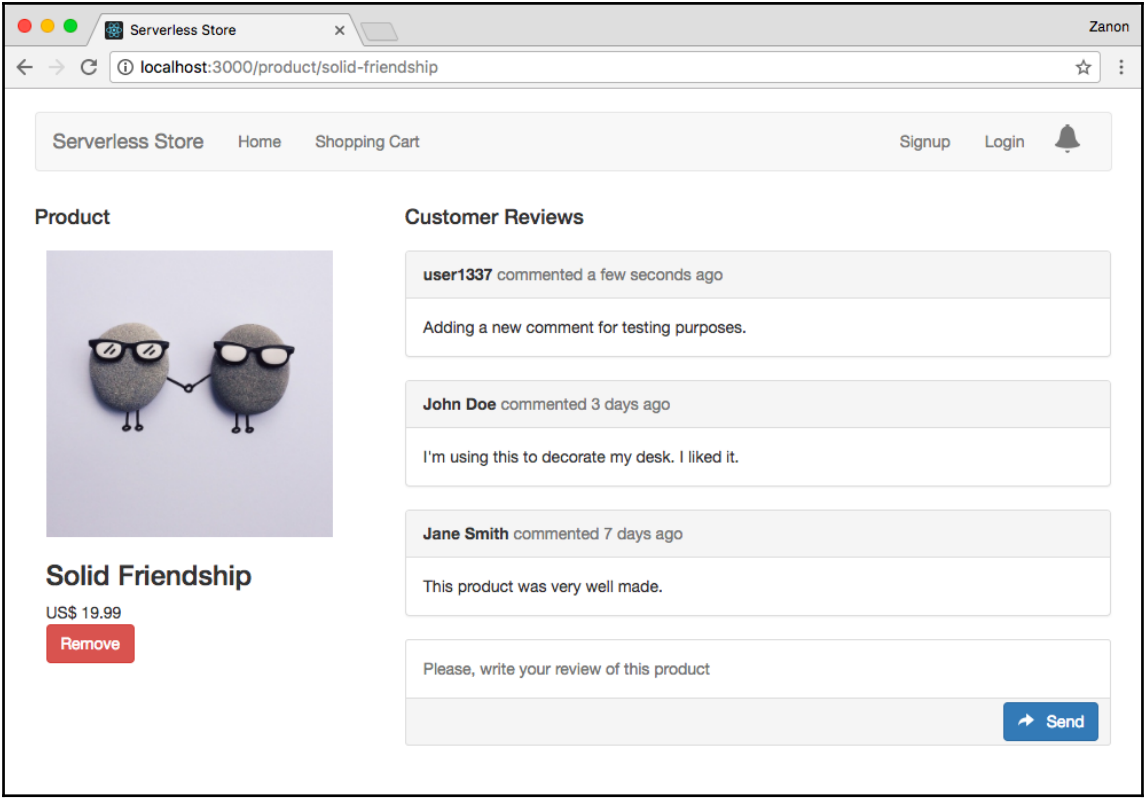
Endpoint

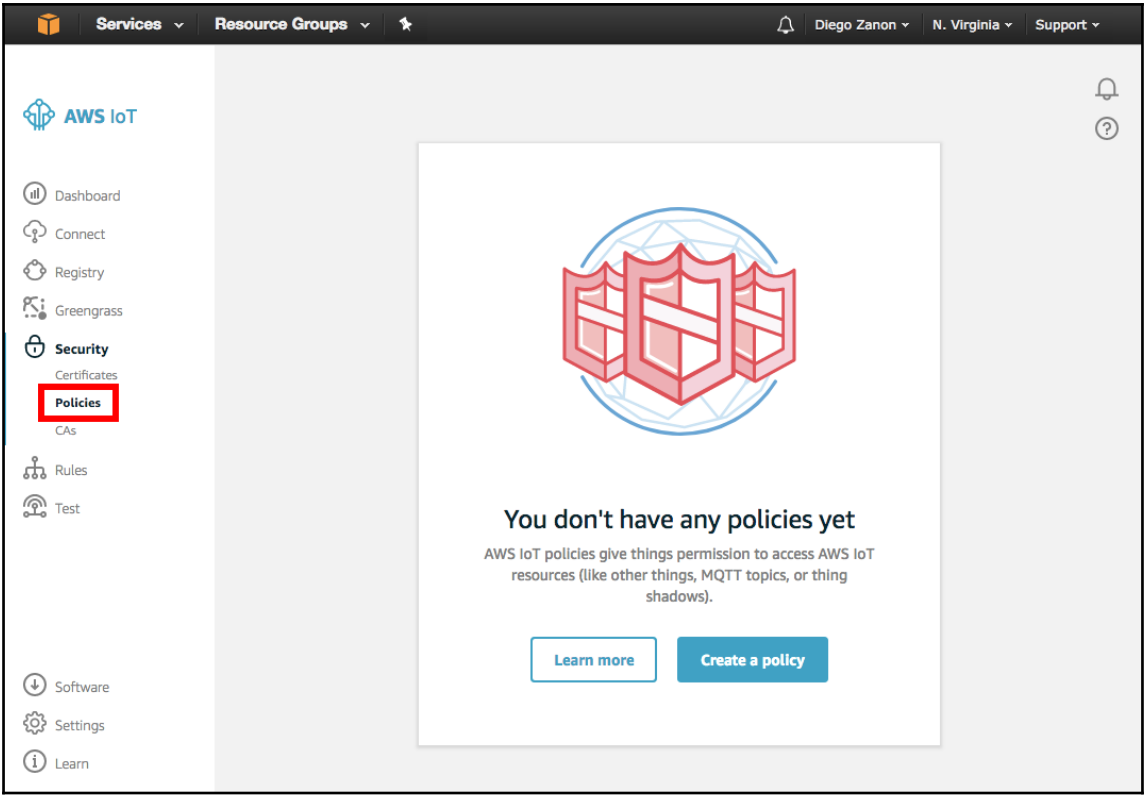
aghmdogbzjnwu.iot.us-east-1.amazonaws.com



Please, write your review of this product

Send





Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters).

Name

iot-policy

Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced mode

Action

iot:Connect, iot:Subscribe, iot:Publish, iot:Receive

Resource ARN

*

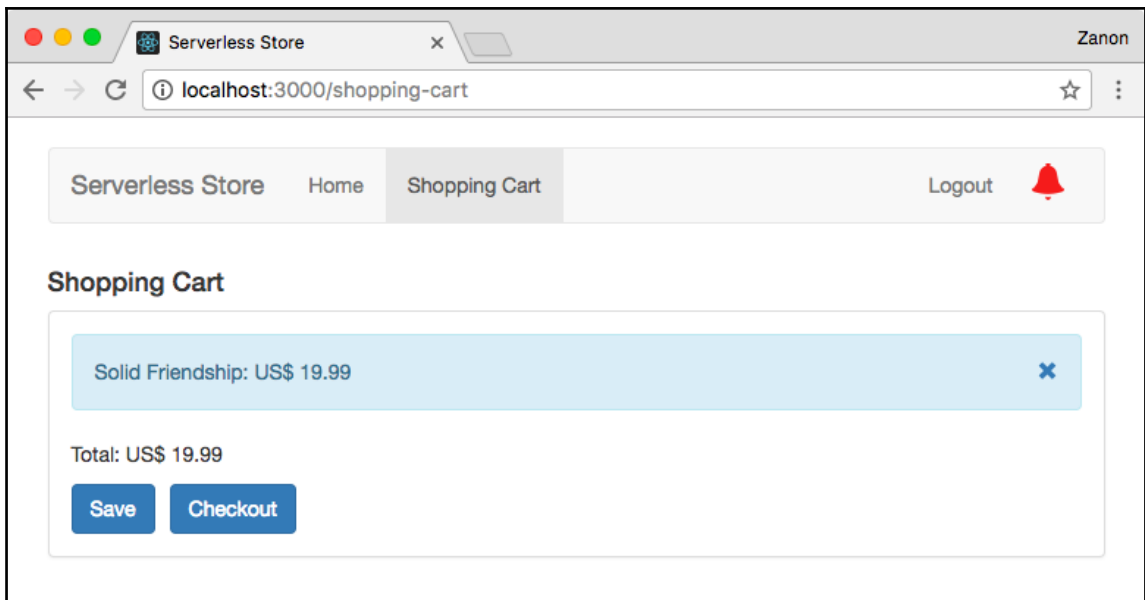
Effect

☒ Allow

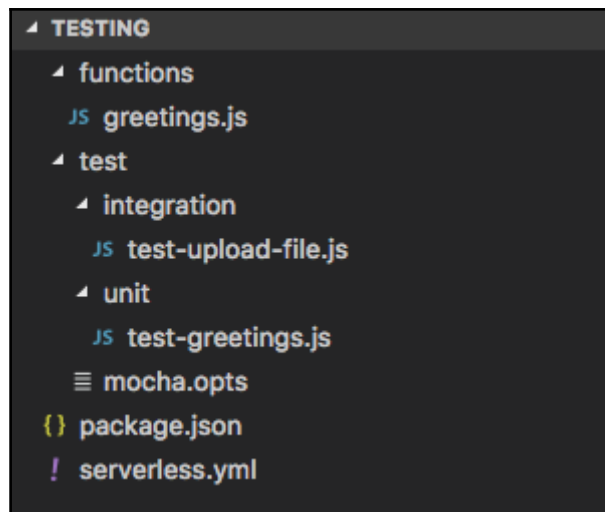
☐ Deny

Remove

Create



Chapter 10: Testing, Deploying, and Monitoring



```
MacBook:backend zanon$ npm test

> testing@1.0.0 test /Users/zanon/Desktop/testing/backend
> mocha

  Greetings
    #hello()
      ✓ should return hello + name

  1 passing (11ms)

MacBook:backend zanon$
```

```
Greetings
  #hello()
    1) should return hello + name

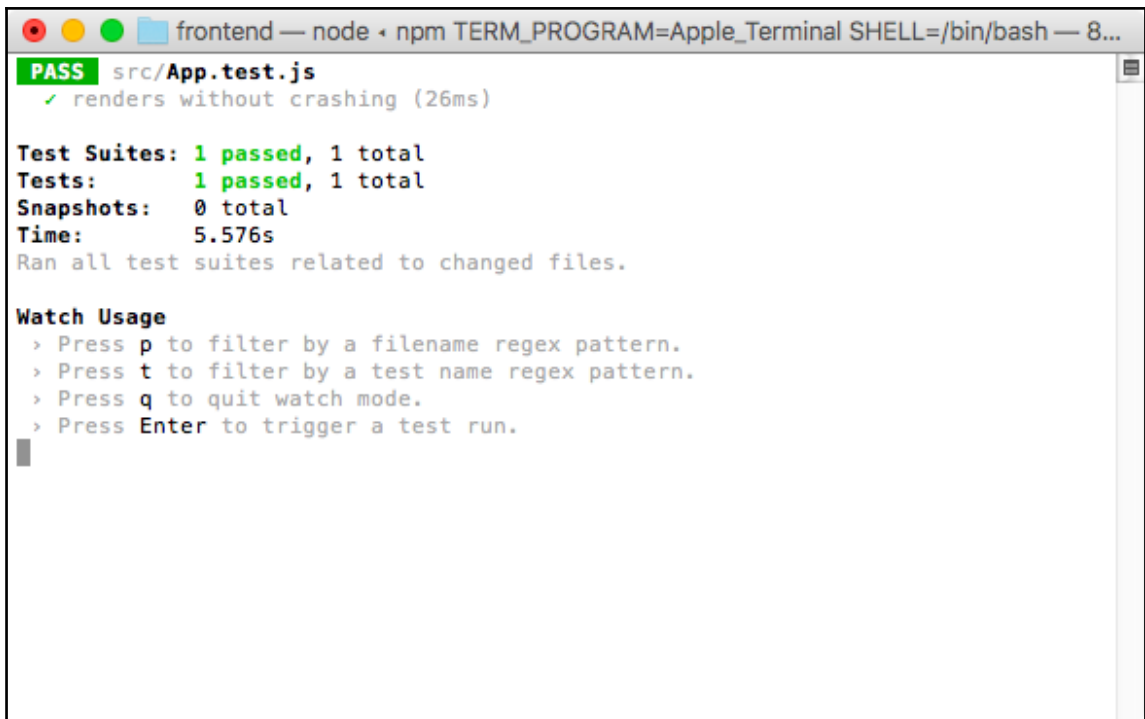
0 passing (15ms)
1 failing

1) Greetings #hello() should return hello + name:

  AssertionError: 'Bye, John!' == 'Hello, John!'
    + expected - actual

    -Bye, John!
    +Hello, John!

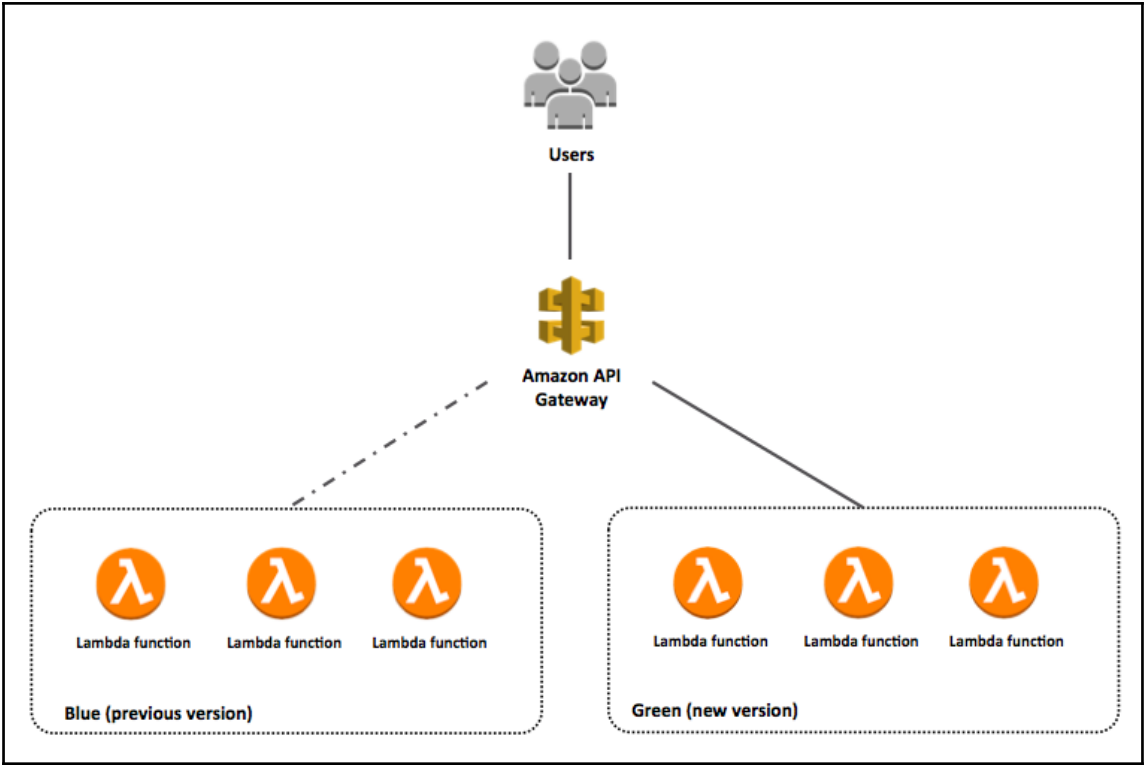
    at greetings.hello (test/unit/test-greetings.js:29:24)
    at Object.module.exports.hello (lib/greetings.js:5:5)
    at Context.it (test/unit/test-greetings.js:23:23)
```

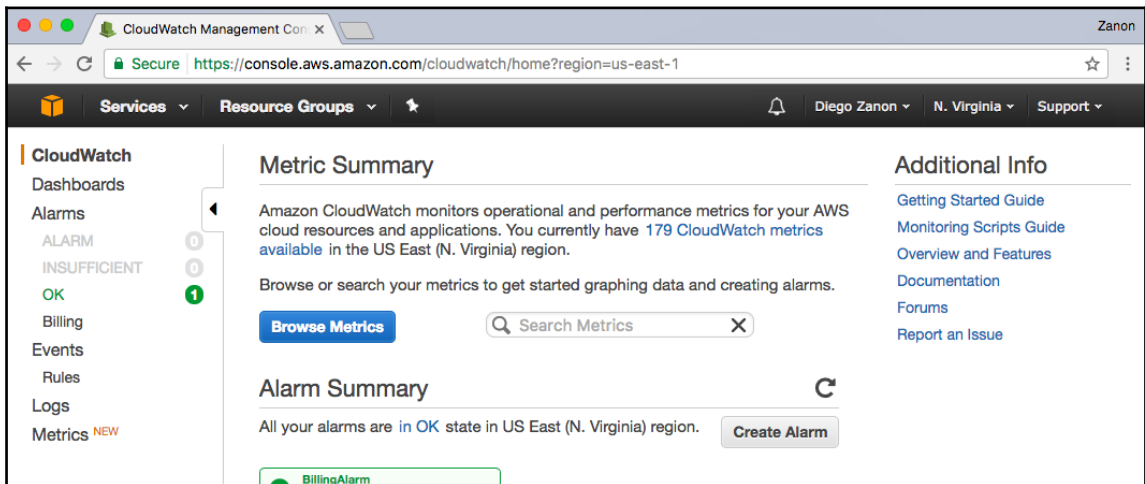
```
frontend — node • npm TERM_PROGRAM=Apple_Terminal SHELL=/bin/bash — 8...
PASS src/App.test.js
  ✓ renders without crashing (26ms)

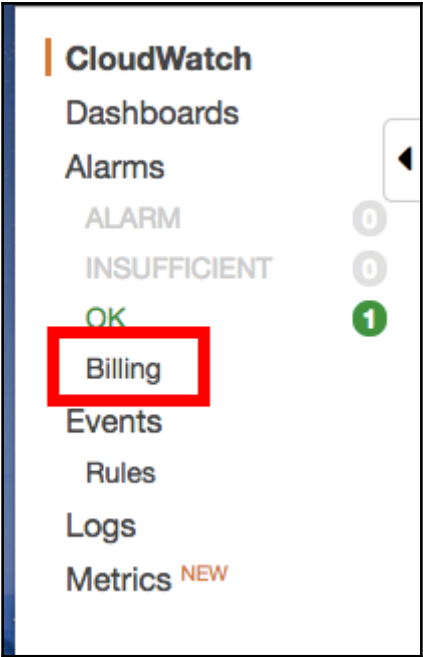
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        5.576s
Ran all test suites related to changed files.

Watch Usage
  > Press p to filter by a filename regex pattern.
  > Press t to filter by a test name regex pattern.
  > Press q to quit watch mode.
  > Press Enter to trigger a test run.
```



```
MacBook:backend zanon$ serverless deploy list
Serverless: Listing deployments:
Serverless: -----
Serverless: Timestamp: 1499216161420
Serverless: Datetime: 2017-07-05T00:56:01.420Z
Serverless: Files:
Serverless: - compiled-cloudformation-template.json
Serverless: - testing.zip
Serverless: -----
Serverless: Timestamp: 1499216616127
Serverless: Datetime: 2017-07-05T01:03:36.127Z
Serverless: Files:
Serverless: - compiled-cloudformation-template.json
Serverless: - testing.zip
Serverless: -----
Serverless: Timestamp: 1499217093742
Serverless: Datetime: 2017-07-05T01:11:33.742Z
Serverless: Files:
Serverless: - compiled-cloudformation-template.json
Serverless: - testing.zip
MacBook:backend zanon$
```





Create Alarm

1. Select Metric

2. Define Alarm

Browse Metrics

Q Search Metrics

X

CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: 179

ApiGateway Metrics : 40

By Api Name : 20

By Stage : 20

Billing Metrics : 14

Total Estimated Charge : 1

By Service : 13

CloudFront Metrics : 30

Per-Distribution Metrics : 30

IoT Metrics : 5

Protocol Metrics : 5

Lambda Metrics : 36

Across All Functions : 4

By Function Name : 16

By Resource : 16

Logs Metrics : 10

Account Metrics : 2

Log Group Metrics : 8

S3 Metrics : 44

Storage Metrics : 44

Update Graph

Cancel

Previous

Next

Create Alarm

Create Alarm

1. Select Metric

2. Define Alarm

Billing

Q AWSLambda EstimatedChar X

1 to 1 of 1 metrics

Total Estimated Charge

By Service

Billing > By Service

ServiceName	Currency	Metric Name
<input type="checkbox"/> AWSLambda	USD	EstimatedCharges

Select a metric above to view graph

Click a checkbox to select a metric

Click on text to add to search

Update Graph

Time Range

RelativeAbsoluteUTC (GMT)

From: 12.08 hours ago

To: 0 hours ago

Zoom: 1h | 3h | 6h | 12h | 1d | 3d | 1w | 2w

Left Y-axis

LimitsMin0Max

AutoAuto

Cancel

Previous

Next

Create Alarm

Create Alarm

1. Select Metric

2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name:

Description:

Whenever charges for:

EstimatedCharges

is:

>=

USD \$

0

Actions

Define what actions are taken when your alarm changes state.

Notification

Delete

Whenever this alarm:

State is ALARM

Send notification to:

Select a notification list

New list

Enter list

+ Notification


+ AutoScaling Action

+ EC2 Action

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line

EstimatedCharges >= 0



Namespace:

AWS/Billing

ServiceName:

AWSLambda

Currency:

USD

Metric Name:

EstimatedCharges

Cancel

Previous

Next

Create Alarm

```
MacBook:monitoring zanon$ serverless metrics
Service wide metrics
July 7, 2017 1:12 AM - July 8, 2017 1:12 AM

Invocations: 6
Throttles: 0
Errors: 1
Duration (avg.): 10.82ms
MacBook:monitoring zanon$
```

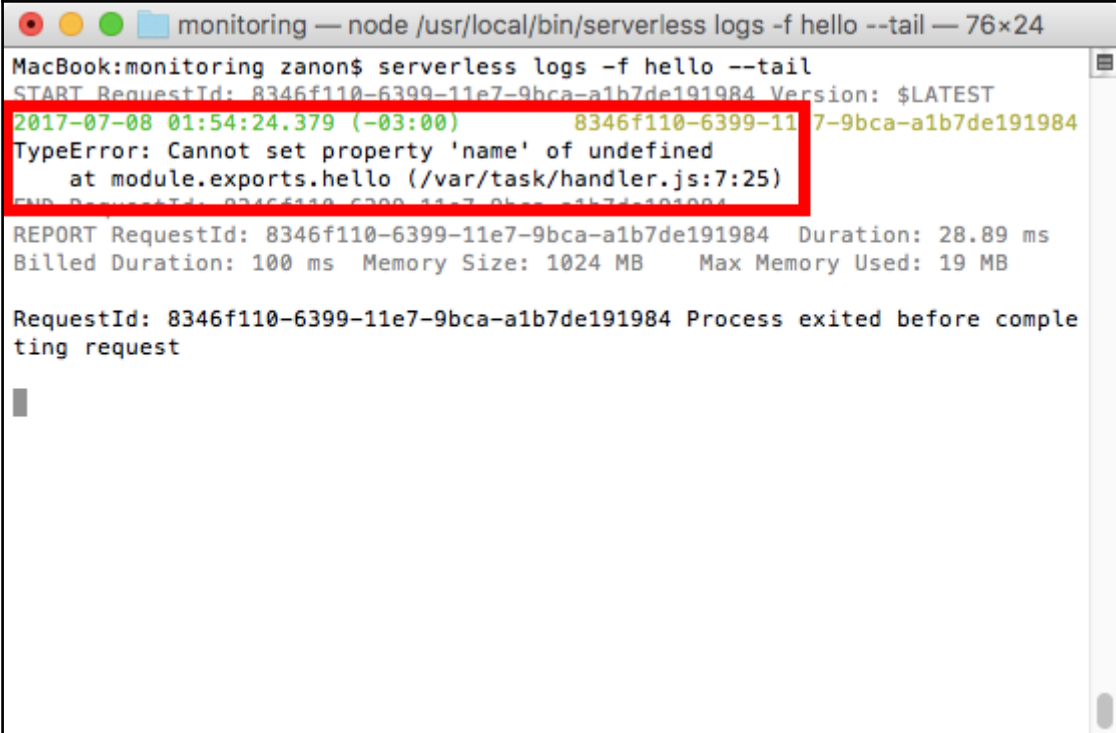
```
MacBook:monitoring zanon$ serverless invoke -f hello
{
  "errorMessage": "RequestId: 0bcf8093-6393-11e7-b4eb-47037b3721d0 Process exited before completing request"
}

Error -----
Invoked function failed

For debugging logs, run again after setting the "SLS_DEBUG=*" environment variable.

Get Support -----
Docs:      docs.serverless.com
Bugs:      github.com/serverless/serverless/issues
Forums:    forum.serverless.com
Chat:      gitter.im/serverless/serverless

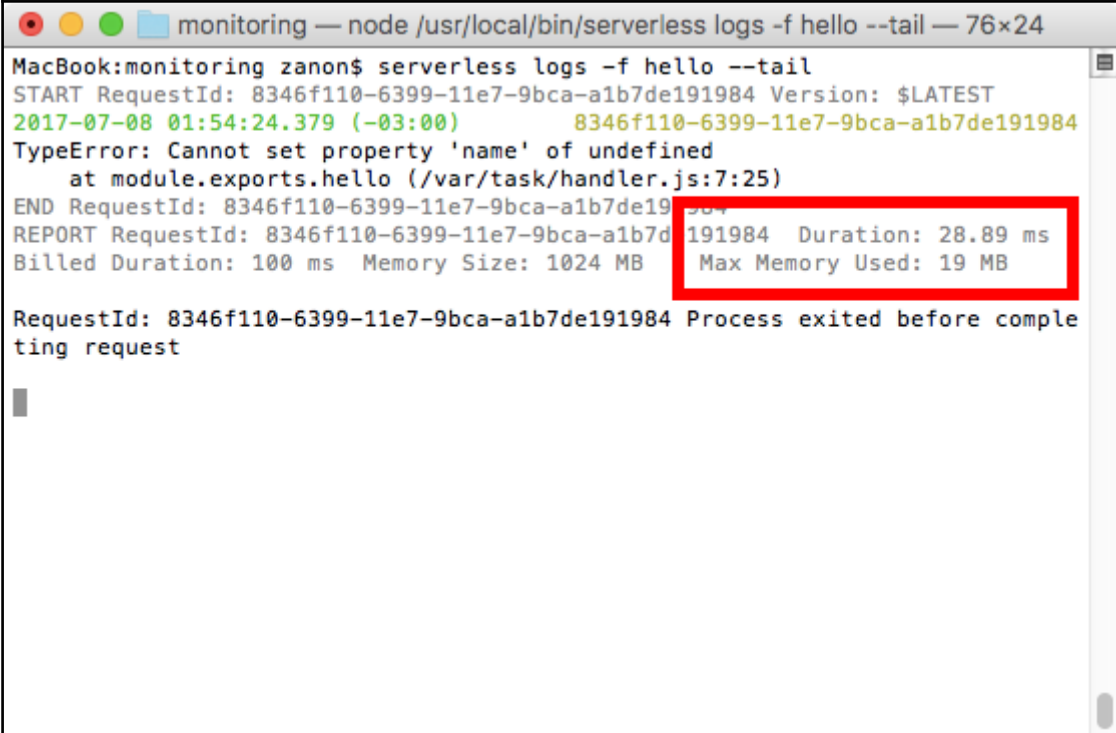
Your Environment Information -----
OS:        darwin
Node Version: 6.10.3
Serverless Version: 1.17.0
MacBook:monitoring zanon$
```

A terminal window titled "monitoring — node /usr/local/bin/serverless logs -f hello --tail — 76x24". The terminal shows the command "MacBook:monitoring zanon\$ serverless logs -f hello --tail". The output includes a "START" line with a RequestId, a timestamp "2017-07-08 01:54:24.379 (-03:00)", and a RequestId. A red box highlights the error message: "TypeError: Cannot set property 'name' of undefined" and "at module.exports.hello (/var/task/handler.js:7:25)". Below the error, there is a "REPORT" line with RequestId, Duration, Billed Duration, Memory Size, and Max Memory Used. The terminal ends with "RequestId: 8346f110-6399-11e7-9bca-a1b7de191984 Process exited before completing request".

```
MacBook:monitoring zanon$ serverless logs -f hello --tail
START RequestId: 8346f110-6399-11e7-9bca-a1b7de191984 Version: $LATEST
2017-07-08 01:54:24.379 (-03:00) 8346f110-6399-11e7-9bca-a1b7de191984
TypeError: Cannot set property 'name' of undefined
    at module.exports.hello (/var/task/handler.js:7:25)
END RequestId: 8346f110-6399-11e7-9bca-a1b7de191984
REPORT RequestId: 8346f110-6399-11e7-9bca-a1b7de191984 Duration: 28.89 ms
Billed Duration: 100 ms Memory Size: 1024 MB Max Memory Used: 19 MB

RequestId: 8346f110-6399-11e7-9bca-a1b7de191984 Process exited before completing request
```



A terminal window titled "monitoring — node /usr/local/bin/serverless logs -f hello --tail — 76x24". The window shows the output of the "serverless logs -f hello --tail" command. The output includes a "START" line with a RequestId, a timestamp, and a version. It then shows a "TypeError: Cannot set property 'name' of undefined" at a specific line in a handler file. Following this, an "END" line is shown. A "REPORT" line provides performance metrics: "Duration: 28.89 ms", "Billed Duration: 100 ms", "Memory Size: 1024 MB", and "Max Memory Used: 19 MB". The last line indicates the "Process exited before completing request". A red rectangular box highlights the "REPORT" line and the "Max Memory Used: 19 MB" value.

```
MacBook:monitoring zanon$ serverless logs -f hello --tail
START RequestId: 8346f110-6399-11e7-9bca-a1b7de191984 Version: $LATEST
2017-07-08 01:54:24.379 (-03:00) 8346f110-6399-11e7-9bca-a1b7de191984
TypeError: Cannot set property 'name' of undefined
    at module.exports.hello (/var/task/handler.js:7:25)
END RequestId: 8346f110-6399-11e7-9bca-a1b7de191984
REPORT RequestId: 8346f110-6399-11e7-9bca-a1b7de191984 Duration: 28.89 ms
Billed Duration: 100 ms Memory Size: 1024 MB Max Memory Used: 19 MB

RequestId: 8346f110-6399-11e7-9bca-a1b7de191984 Process exited before comple
ting request
```