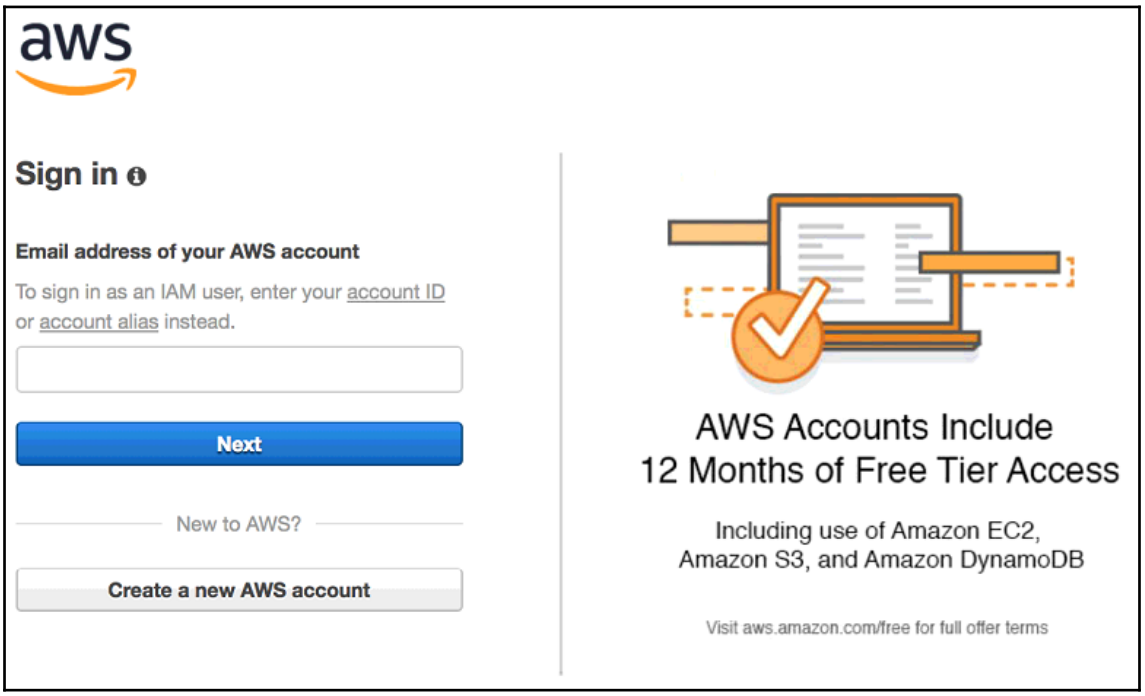
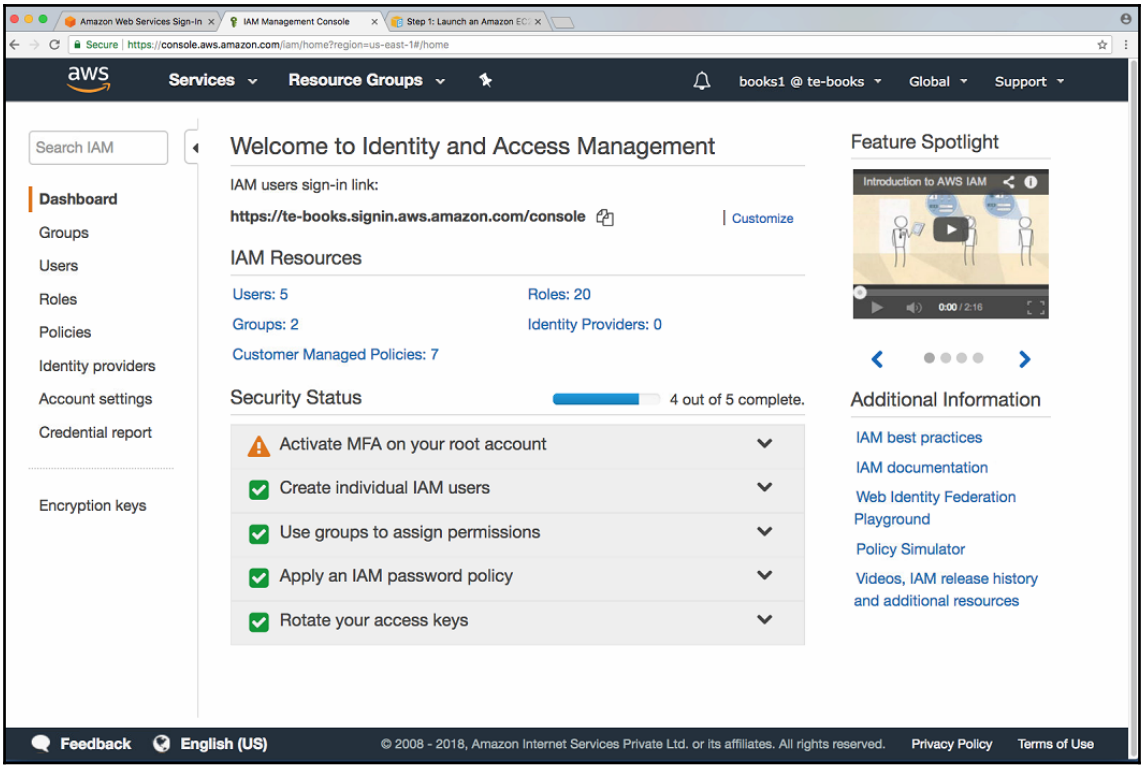
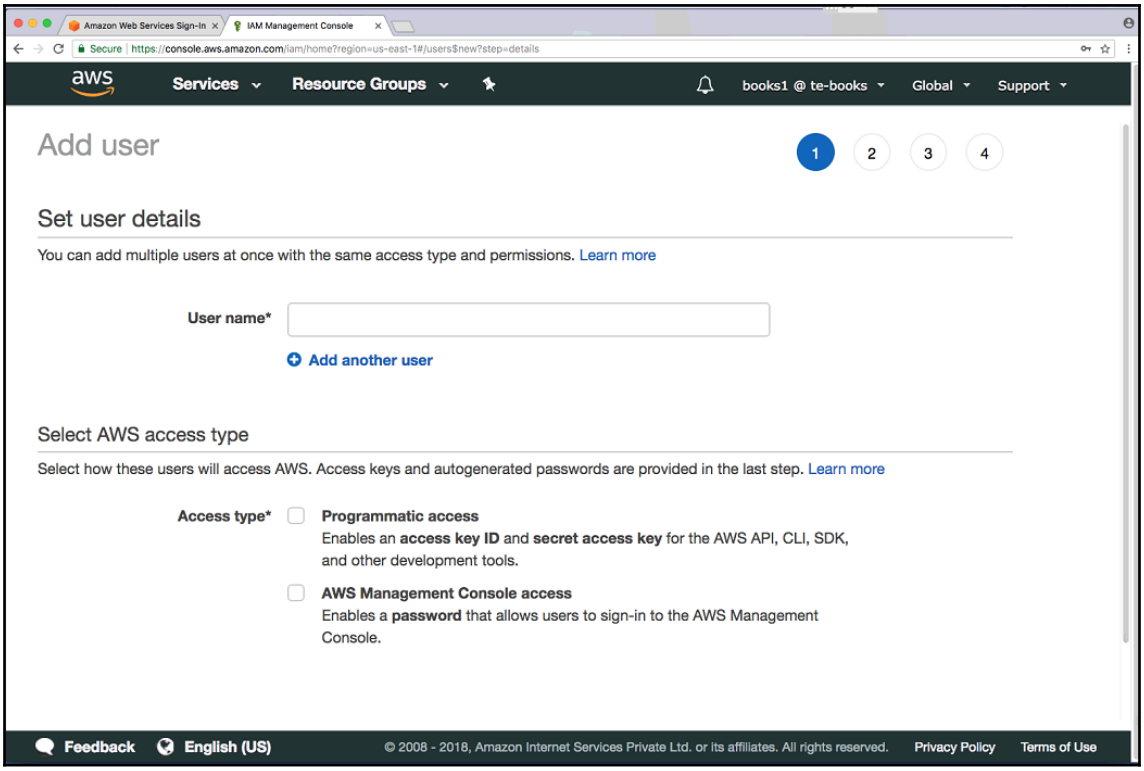
Chapter 1: Getting Started with AWS

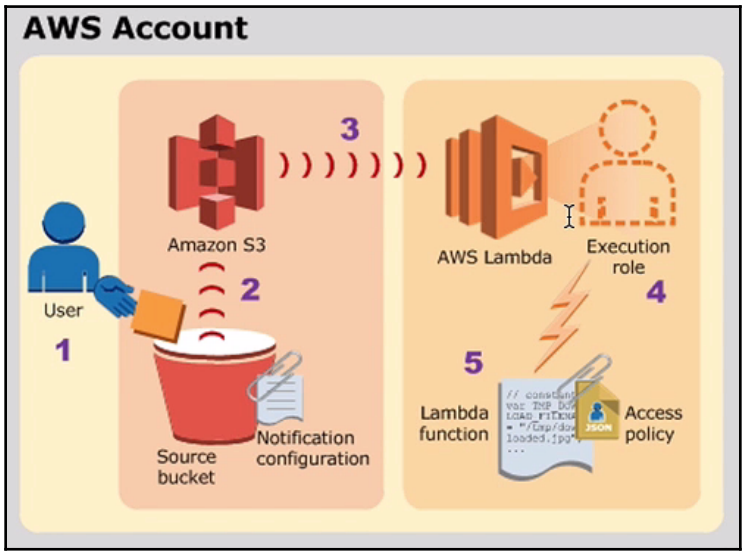


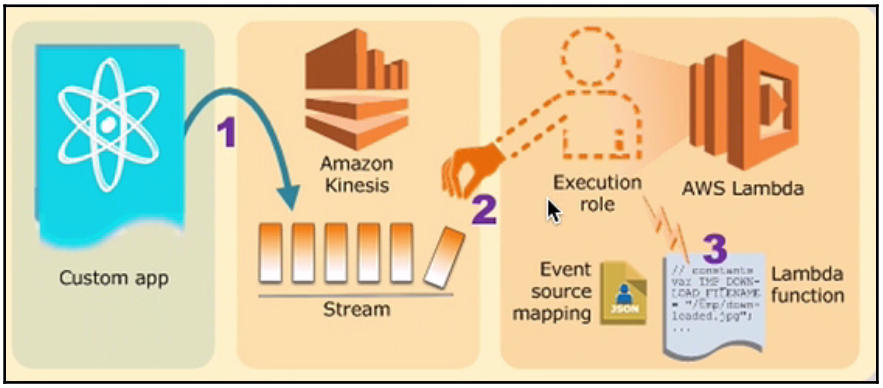

Amazon Web Services Sign-In
IAM Management Console
Step 1: Launch an Amazon EC
Secure | https://console.aws.amazon.com/iam/home?region=us-east-1#/home
aws
Services
Resource Groups
books1 @ te-books
Global
Support
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Welcome to Identity and Access Management
IAM users sign-in link:
https://te-books.signin.aws.amazon.com/console
Customize
IAM Resources
Users: 5
Roles: 20
Groups: 2
Identity Providers: 0
Customer Managed Policies: 7
Security Status
4 out of 5 complete.
Activate MFA on your root account
Create individual IAM users
Use groups to assign permissions
Apply an IAM password policy
Rotate your access keys
Feature Spotlight
Introduction to AWS IAM
0:00 / 2:16
Additional Information
IAM best practices
IAM documentation
Web Identity Federation Playground
Policy Simulator
Videos, IAM release history and additional resources
Feedback
English (US)
© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Amazon Web Services Sign-In
IAM Management Console
Secure | https://console.aws.amazon.com/iam/home?region=us-east-1#/users$new?step=details
aws
Services
Resource Groups
books1 @ te-books
Global
Support
Add user
1
2
3
4
Set user details
You can add multiple users at once with the same access type and permissions. Learn more
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.
Feedback
English (US)
© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Secure https://aws.amazon.com/lambda/
AWS Lambda
Run code without thinking about servers.
Pay for only the compute time you consume.
Get started with AWS Lambda
Product Details
Pricing
Getting Started
Partners
FAQs
Documentation

- Building Applications with AWS Lambda
  - **Event Source Mapping**
  - Supported Event Sources
  - Deploying Lambda-based Applications





Monthly compute charges

The monthly compute price is $0.00001667 per GB-s and the free tier provides 400,000 GB-s.

Total compute (seconds) = 3M * (1s) = 3,000,000 seconds

Total compute (GB-s) = 3,000,000 * 512MB/1024 = 1,500,000 GB-s

Total compute – Free tier compute = Monthly billable compute GB- s

1,500,000 GB-s – 400,000 free tier GB-s = 1,100,000 GB-s

Monthly compute charges = 1,100,000 * $0.00001667 = $18.34

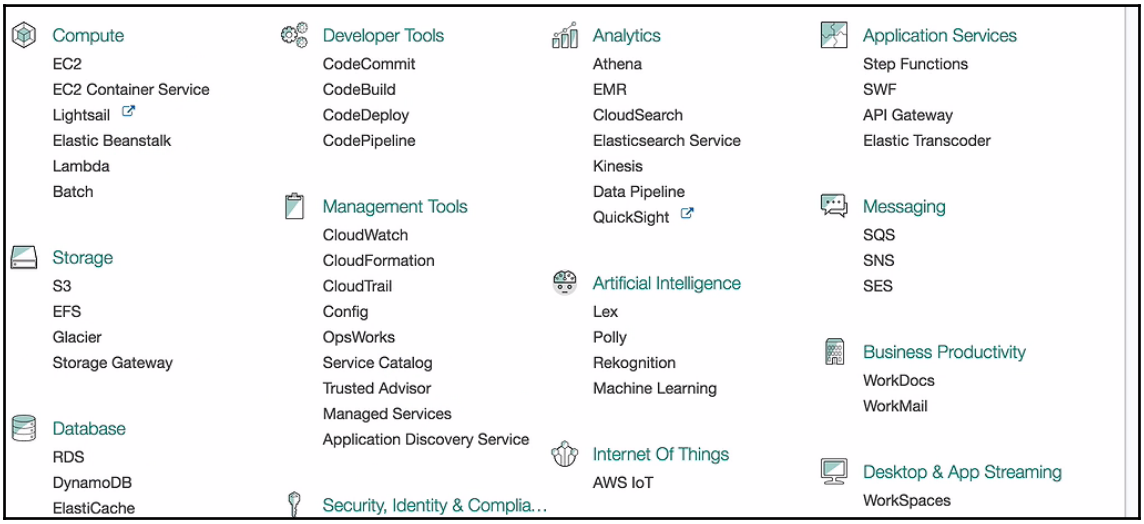
Monthly request charges

The monthly request price is $0.20 per 1 million requests and the free tier provides 1M requests per month.

Total requests – Free tier requests = Monthly billable requests

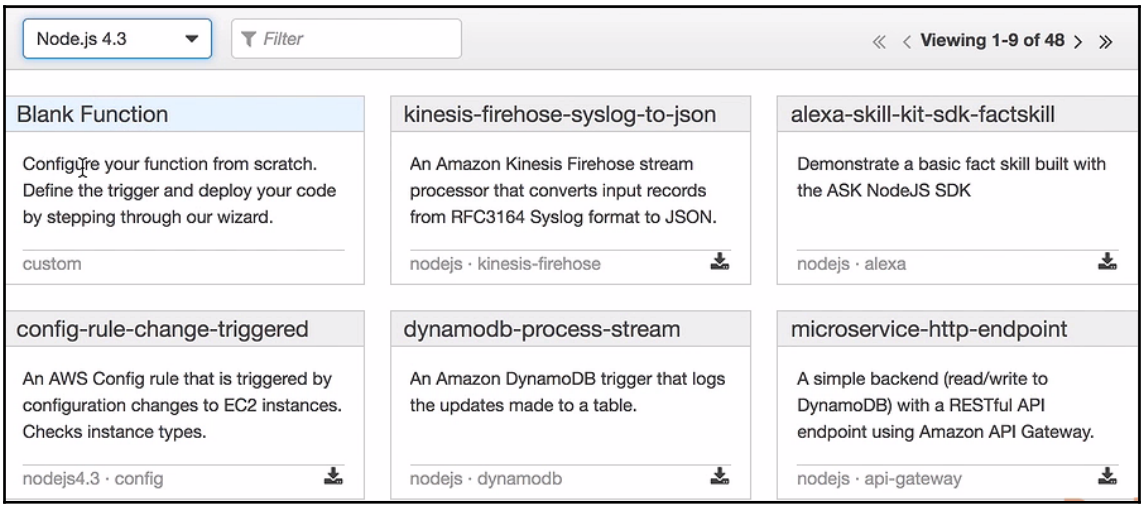
3M requests – 1M free tier requests = 2M Monthly billable requests

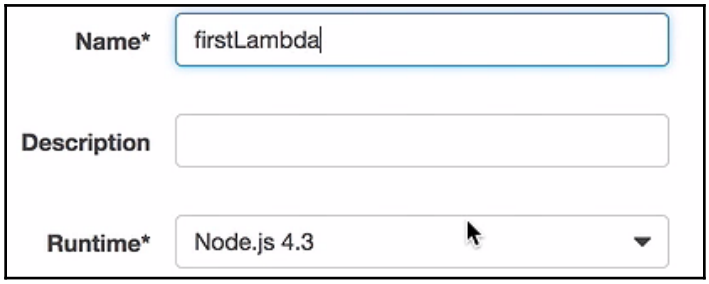
Monthly request charges = 2M * $0.2/M = $0.40

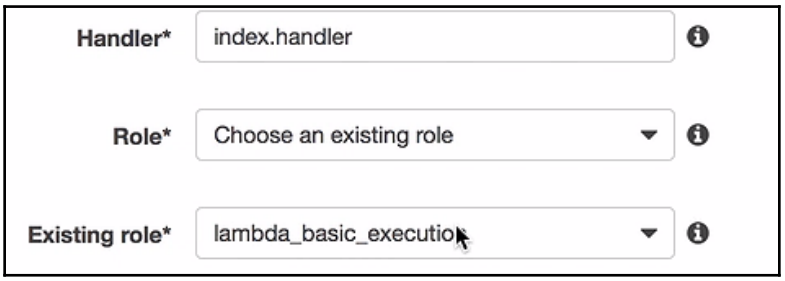


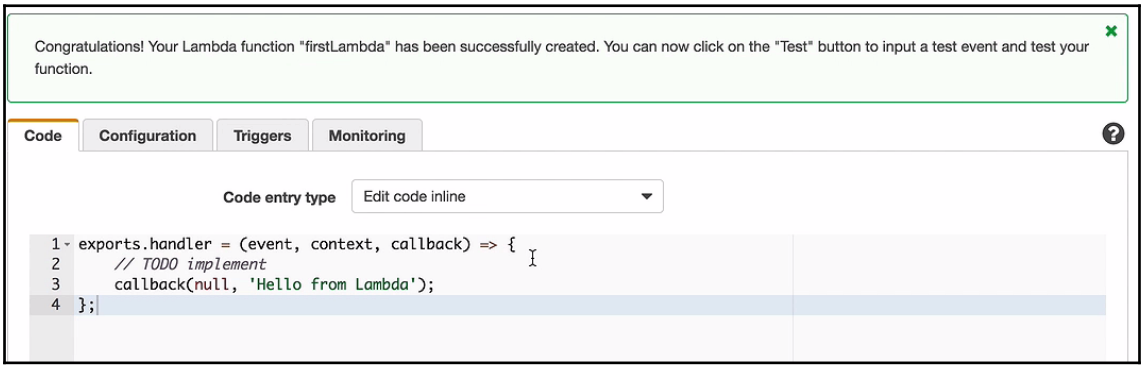
AWS Lambda

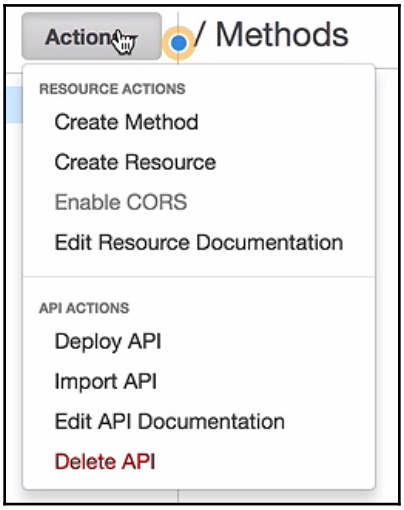
Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	lambda.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	lambda.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	lambda.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	lambda.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	lambda.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	lambda.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	lambda.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	lambda.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	lambda.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	lambda.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	lambda.eu-west-2.amazonaws.com	HTTPS

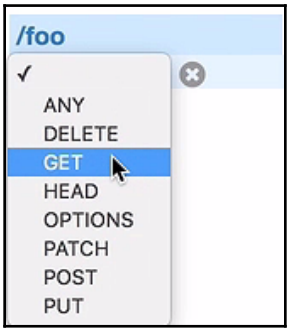


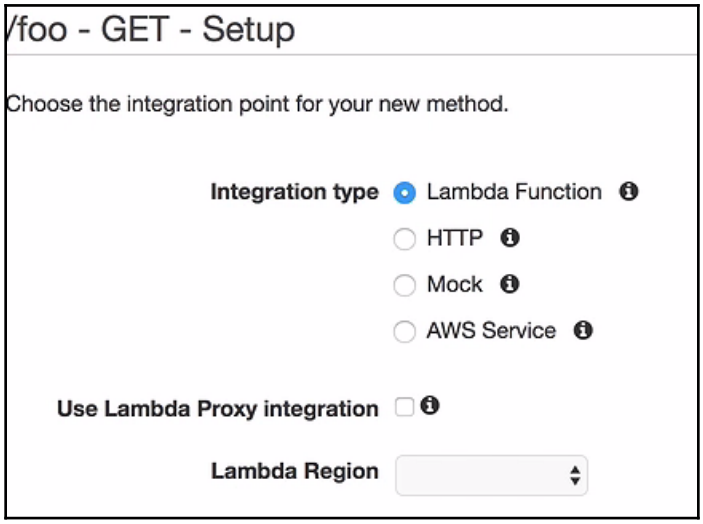

Name*
firstLambda
Description
Runtime*
Node.js 4.3

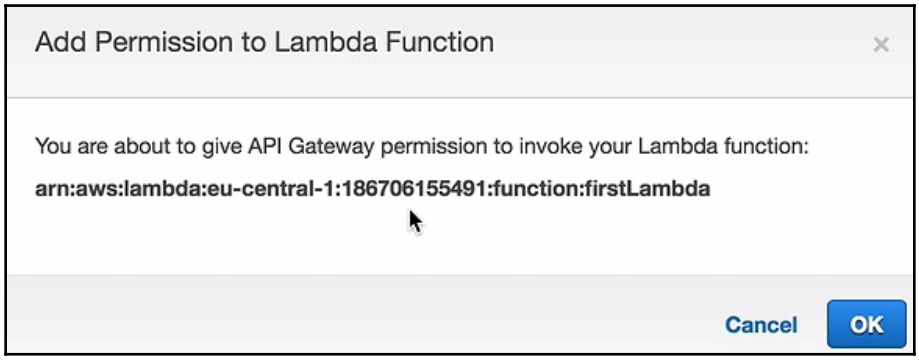

Handler*
index.handler
Role*
Choose an existing role
Existing role*
lambda_basic_execution

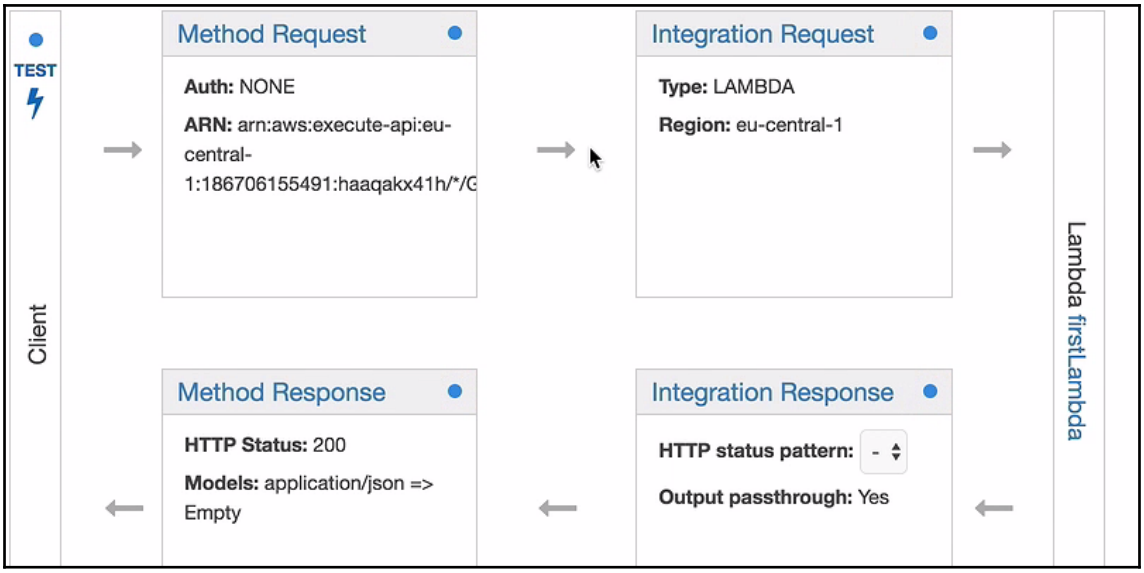

Congratulations! Your Lambda function "firstLambda" has been successfully created. You can now click on the "Test" button to input a test event and test your function.
Code
Configuration
Triggers
Monitoring
Code entry type
Edit code inline
1 exports.handler = (event, context, callback) => {
2 // TODO implement
3 callback(null, 'Hello from Lambda');
4 };


Actions
/ Methods
RESOURCE ACTIONS
Create Method
Create Resource
Enable CORS
Edit Resource Documentation
API ACTIONS
Deploy API
Import API
Edit API Documentation
Delete API


/foo
✓
ANY
DELETE
GET
HEAD
OPTIONS
PATCH
POST
PUT


/foo - GET - Setup
Choose the integration point for your new method.
Integration type
Lambda Function
HTTP
Mock
AWS Service
Use Lambda Proxy integration
Lambda Region


Add Permission to Lambda Function
You are about to give API Gateway permission to invoke your Lambda function:
arn:aws:lambda:eu-central-1:186706155491:function:firstLambda
Cancel
OK


TEST
Client
Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-central-1:186706155491:haaqakx41h/*/G
Integration Request
Type: LAMBDA
Region: eu-central-1
Lambda firstLambda
Method Response
HTTP Status: 200
Models: application/json => Empty
Integration Response
HTTP status pattern: -
Output passthrough: Yes

Request: /foo

Status: 200

Latency: 90 ms

Response Body

```
"Hello from Lambda"
```

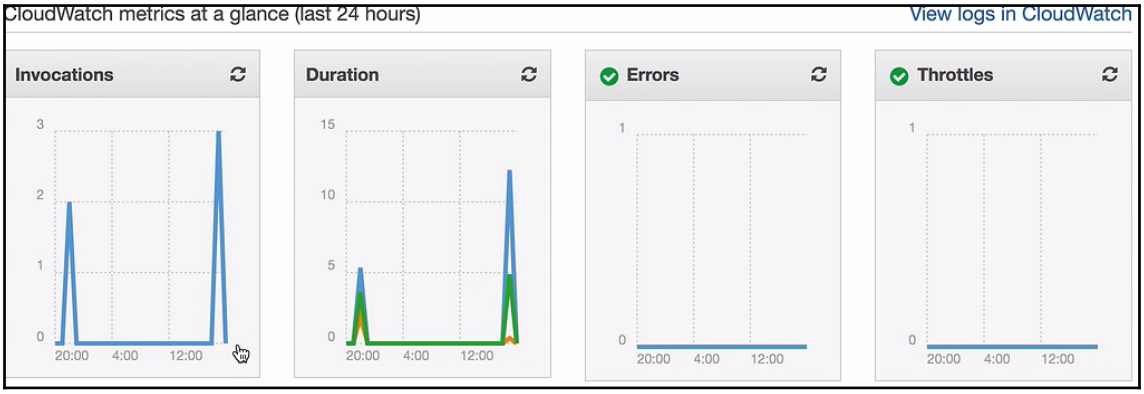
Response Headers

```
{"X-Amzn-Trace-Id":"Root=1-589a16ee-45e546c29c2a9
6e41d65979b","Content-Type":"application/json"}
```

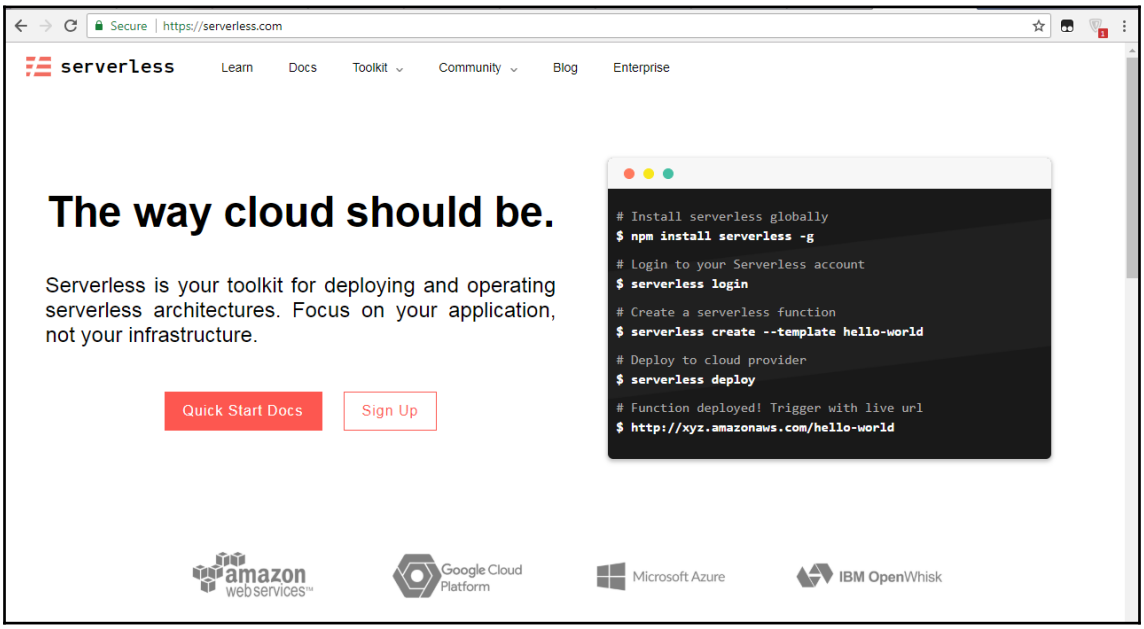
Logs

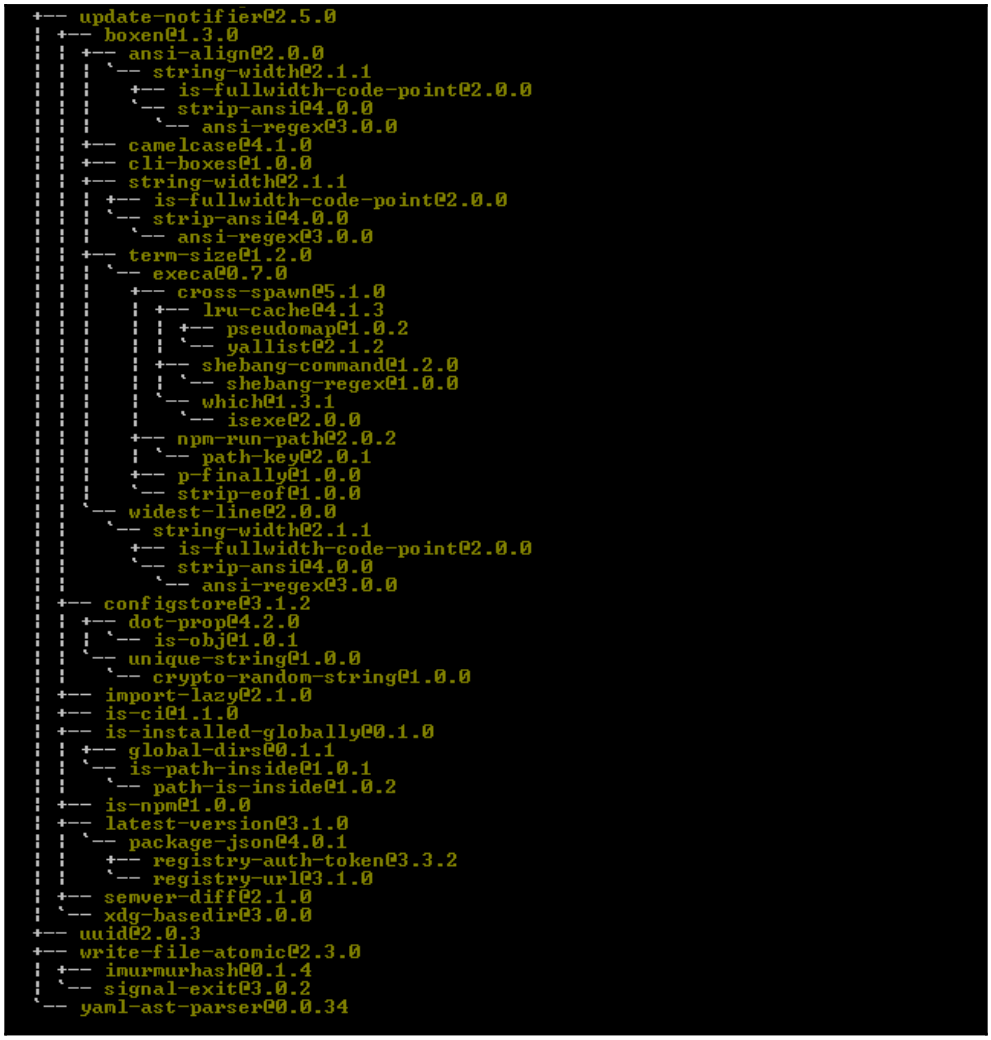
```
Execution log for request test-request
Tue Feb 07 18:50:22 UTC 2017 : Starting execution
for request: test-invoke-request
Tue Feb 07 18:50:22 UTC 2017 : HTTP Method: GET,
 Resource Path: /foo
Tue Feb 07 18:50:22 UTC 2017 : Method request pat
h: {}
Tue Feb 07 18:50:22 UTC 2017 : Method request que
ry string: {}
Tue Feb 07 18:50:22 UTC 2017 : Method request hea
ders: {}
Tue Feb 07 18:50:22 UTC 2017 : Method request bod
y before transformations:
```

CloudWatch metrics at a glance (last 24 hours) View logs in CloudWatch



Chapter 2:
Exploring the Serverless Framework




+-- update-notifier@2.5.0
| +-- boxen@1.3.0
| | +-- ansi-align@2.0.0
| | | `-- string-width@2.1.1
| | | +-- is-fullwidth-code-point@2.0.0
| | | `-- strip-ansi@4.0.0
| | | `-- ansi-regex@3.0.0
| | +-- camelcase@4.1.0
| | +-- cli-boxes@1.0.0
| | +-- string-width@2.1.1
| | | +-- is-fullwidth-code-point@2.0.0
| | | `-- strip-ansi@4.0.0
| | | `-- ansi-regex@3.0.0
| | +-- term-size@1.2.0
| | | `-- execa@0.7.0
| | | +-- cross-spawn@5.1.0
| | | | +-- lru-cache@4.1.3
| | | | | +-- pseudomap@1.0.2
| | | | | `-- yallist@2.1.2
| | | | +-- shebang-command@1.2.0
| | | | | `-- shebang-regex@1.0.0
| | | | `-- which@1.3.1
| | | | `-- isexe@2.0.0
| | | +-- npm-run-path@2.0.2
| | | | `-- path-key@2.0.1
| | | +-- p-finally@1.0.0
| | | `-- strip-eof@1.0.0
| | `-- widest-line@2.0.0
| | `-- string-width@2.1.1
| | +-- is-fullwidth-code-point@2.0.0
| | `-- strip-ansi@4.0.0
| | `-- ansi-regex@3.0.0
| +-- configstore@3.1.2
| | +-- dot-prop@4.2.0
| | | `-- is-obj@1.0.1
| | `-- unique-string@1.0.0
| | `-- crypto-random-string@1.0.0
| +-- import-lazy@2.1.0
| +-- is-ci@1.1.0
| +-- is-installed-globally@0.1.0
| | +-- global-dirs@0.1.1
| | `-- is-path-inside@1.0.1
| | `-- path-is-inside@1.0.2
| +-- is-npm@1.0.0
| +-- latest-version@3.1.0
| | `-- package-json@4.0.1
| | +-- registry-auth-token@3.3.2
| | `-- registry-url@3.1.0
| +-- semver-diff@2.1.0
| `-- xdg-basedir@3.0.0
+-- uuid@2.0.3
+-- write-file-atomic@2.3.0
| +-- imurmurhash@0.1.4
| `-- signal-exit@3.0.2
`-- yaml-ast-parser@0.0.34

```
Admin@Admin:~/Documents/workspace-vscode-1.17.2/microservices-orders$ serverless

Commands
* You can run commands with "serverless" or the shortcut "sls"
* Pass "--verbose" to this command to get in-depth plugin info
* Pass "--no-color" to disable CLI colors
* Pass "--help" after any <command> for contextual help

Framework
* Documentation: https://serverless.com/framework/docs/

config ..................... Configure Serverless
config credentials ......... Configures a new provider profile for the Serverless Framework
create ..................... Create new Serverless service
create_domain .............. Creates a domain using the domain name defined in the serverless file
delete_domain .............. Deletes a domain using the domain name defined in the serverless file
deploy ..................... Deploy a Serverless service
deploy function ............ Deploy a single function from the service
deploy list ................ List deployed version of your Serverless Service
deploy list functions ...... List all the deployed functions and their versions
info ....................... Display information about the service
install .................... Install a Serverless service from GitHub or a plugin from the Serverless registry
invoke ..................... Invoke a deployed function
invoke local ............... Invoke function locally
logs ....................... Output the logs of a deployed function
metrics .................... Show metrics for a specific function
offline .................... Simulates API Gateway to call your lambda functions offline.
offline start .............. Simulates API Gateway to call your lambda functions offline using backward compatible initialization.
package .................... Packages a Serverless service
plugin ..................... Plugin management for Serverless
plugin install ............. Install and add a plugin to your service
plugin uninstall ........... Uninstall and remove a plugin from your service
plugin list ................ Lists all available plugins
plugin search .............. Search for plugins
print ...................... Print your compiled and resolved config file
remove ..................... Remove Serverless service and all resources
rollback ................... Rollback the Serverless service to a specific deployment
rollback function .......... Rollback the function to a specific version
slstats .................... Enable or disable stats
webpack .................... Bundle with Webpack

Platform (Beta)
* The Serverless Platform is currently in experimental beta. Follow the docs below to get started.
* Documentation: https://serverless.com/platform/docs/

emit ....................... Emits an event to a running Event Gateway
login ...................... Login or sign up for the Serverless Platform
logout ..................... Logout from the Serverless Platform
run ........................ Runs the Event Gateway and the Emulator

Plugins
AwsCommon, AwsCompileAlexaSkillEvents, AwsCompileAlexaSmartHomeEvents, AwsCompileApigEvents, AwsCompileCloudWatchEventEvents, AwsCompileCloudWatchLogEvents, AwsCompileCognitoUserPoolEv
```

```
Admin@Admin:~/Documents/microservices-orders$ sls create --help
Plugin: Create
create ..................... Create new Serverless service
    --template / -t ........... Template for the service. Available templates: "aws-nodejs", "aws-nodejs-typescript", "aws-nodejs-ecma-script", "aws-python", "aws-python3", "aws-groovy-gradle", "aws-java-maven", "aws-java-gradle", "aws-kotlin-jvm-maven", "aws-kotlin-jvm-gradle", "aws-kotlin-nodejs-gradle", "aws-scala-sbt", "aws-csharp", "aws-fsharp", "aws-go", "aws-go-dep", "azure-nodejs", "fn-nodejs", "fn-go", "google-nodejs", "kubeless-python", "kubeless-nodejs", "openwhisk-java-maven", "openwhisk-nodejs", "openwhisk-php", "openwhisk-python", "openwhisk-swift", "spotinst-nodejs", "spotinst-python", "spotinst-ruby", "spotinst-java8", "webtasks-nodejs", "plugin" and "hello-world"
    --template-url / -u ....... Template URL for the service. Supports: GitHub, BitBucket
    --template-path ........... Template local path for the service.
    --path / -p ............... The path where the service should be created (e.g. --path my-service)
    --name / -n ............... Name for the service. Overwrites the default name of the created service.
```

```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>sls create -t aws-nodejs -n blog
Serverless: Generating boilerplate...
 _______                             __
|   _   .-----.----.--.--.-----.----|  .-----.-----.-----.
|   |___|  -__|   _|  |  |  -__|   _|  |  -__|__ --|__ --|
|____   |_____|__|  \___/|_____|__| |__|_____|_____|_____|
|   |   |             The Serverless Application Framework
|       |                           serverless.com, v1.27.3
 -------'

Serverless: Successfully generated boilerplate for template: "aws-nodejs"
```

```
Service Information
service: blog
stage: dev
region: eu-central-1
stack: blog-dev
api keys:
  None
endpoints:
  POST - https://58pletzc5d.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://58pletzc5d.execute-api.eu-central-1.amazonaws.com/dev/articles/{id}
  PUT - https://58pletzc5d.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://58pletzc5d.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://58pletzc5d.execute-api.eu-central-1.amazonaws.com/dev/articles
functions:
  hello: blog-dev-hello
  createArticle: blog-dev-createArticle
  readOneArticle: blog-dev-readOneArticle
  updateArticle: blog-dev-updateArticle
  deleteArticle: blog-dev-deleteArticle
  readAllArticles: blog-dev-readAllArticles
```

```
Service Information
service: simple
stage: dev
region: eu-central-1
stack: simple-dev
api keys:
  None
endpoints:
  GET - https://iqknh40tnh.execute-api.eu-central-1.amazonaws.com/dev/hello
  POST - https://iqknh40tnh.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://iqknh40tnh.execute-api.eu-central-1.amazonaws.com/dev/articles/{id}
  PUT - https://iqknh40tnh.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://iqknh40tnh.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://iqknh40tnh.execute-api.eu-central-1.amazonaws.com/dev/articles
functions:
  hello: simple-dev-hello
```


```
{"message":"Hello World","input":{"resource":"/hello","path":"/hello","httpMethod":"GET","headers":{"Accept":"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8","Accept-Encoding":"gzip, deflate, br","Accept-Language":"en-US,en;q=0.9","CloudFront-Forwarded-Proto":"https","CloudFront-Is-Desktop-Viewer":"true","CloudFront-Is-Mobile-Viewer":"false","CloudFront-Is-SmartTV-Viewer":"false","CloudFront-Is-Tablet-Viewer":"false","CloudFront-Viewer-Country":"IN","Host":"iqknh40tnh.execute-api.eu-central-1.amazonaws.com","upgrade-insecure-requests":"1","User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.79 Safari/537.36","Via":"2.0 7b637ad78977549938b508611f792222.cloudfront.net (CloudFront)","X-Amz-Cf-Id":"aDkMQjTWM4YUsqN2pgJM7yaWbZRR8h9dLXAVEQNEHp232kuMg_e12w==","X-Amzn-Trace-Id":"Root=1-5b1e565e-6fc204daf5dfd42e2f8ab61e","X-Forwarded-For":"182.76.16.118, 54.182.242.125","X-Forwarded-Port":"443","X-Forwarded-Proto":"https"},"queryStringParameters":null,"pathParameters":null,"stageVariables":null,"requestContext":{"resourceId":"1wtbv5","resourcePath":"/hello","httpMethod":"GET","extendedRequestId":"IUJuyE8zFiAFhKg=","requestTime":"11/Jun/2018:11:00:46 +0000","path":"/dev/hello","accountId":"019859648260","protocol":"HTTP/1.1","stage":"dev","requestTimeEpoch":1528714846676,"requestId":"b17eaa27-6d66-11e8-bfd6-7b4d8875a8b7","identity":{"cognitoIdentityPoolId":null,"accountId":null,"cognitoIdentityId":null,"caller":null,"sourceIp":"182.76.16.118","accessKey":null,"cognitoAuthenticationType":null,"cognitoAuthenticationProvider":null,"userArn":null,"userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.79 Safari/537.36","user":null},"apiId":"iqknh40tnh"},"body":null,"isBase64Encoded":false}}
```

```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>sls invoke local -f hello
event is
context is { awsRequestId: 'id',
  invokeid: 'id',
  logGroupName: '/aws/lambda/simple-dev-hello',
  logStreamName: '2015/09/22/[HEAD]13370a84ca4ed8b77c427af260',
  functionVersion: 'HEAD',
  isDefaultFunctionVersion: true,
  functionName: 'simple-dev-hello',
  memoryLimitInMB: '1024',
  succeed: [Function: succeed],
  fail: [Function: fail],
  done: [Function: done],
  getRemainingTimeInMillis: [Function: getRemainingTimeInMillis] }
```

```
{
    "statusCode": 200,
    "ev": "",
    "rt": 6000,
    "fn": "simple-dev-hello",
    "aid": "id"
}
```

```
region: eu-central-1
stack: simple-dev
api keys:
  None
endpoints:
  GET - https://nb2gqgav6i.execute-api.eu-central-1.amazonaws.com/dev/hello
  POST - https://nb2gqgav6i.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://nb2gqgav6i.execute-api.eu-central-1.amazonaws.com/dev/articles/{id}
  PUT - https://nb2gqgav6i.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://nb2gqgav6i.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://nb2gqgav6i.execute-api.eu-central-1.amazonaws.com/dev/articles
functions:
  hello: simple-dev-hello
  createArticle: simple-dev-createArticle
  readOneArticle: simple-dev-readOneArticle
  updateArticle: simple-dev-updateArticle
  deleteArticle: simple-dev-deleteArticle
  readAllArticles: simple-dev-readAllArticles
```

```
{
    "statusCode": 200,
    "ev": {},
    "rt": 5998,
    "fn": "simple-dev-hello",
    "aid": "ca5f1156-ef0d-11e6-bdf4-df482d5dd82e"
}
```

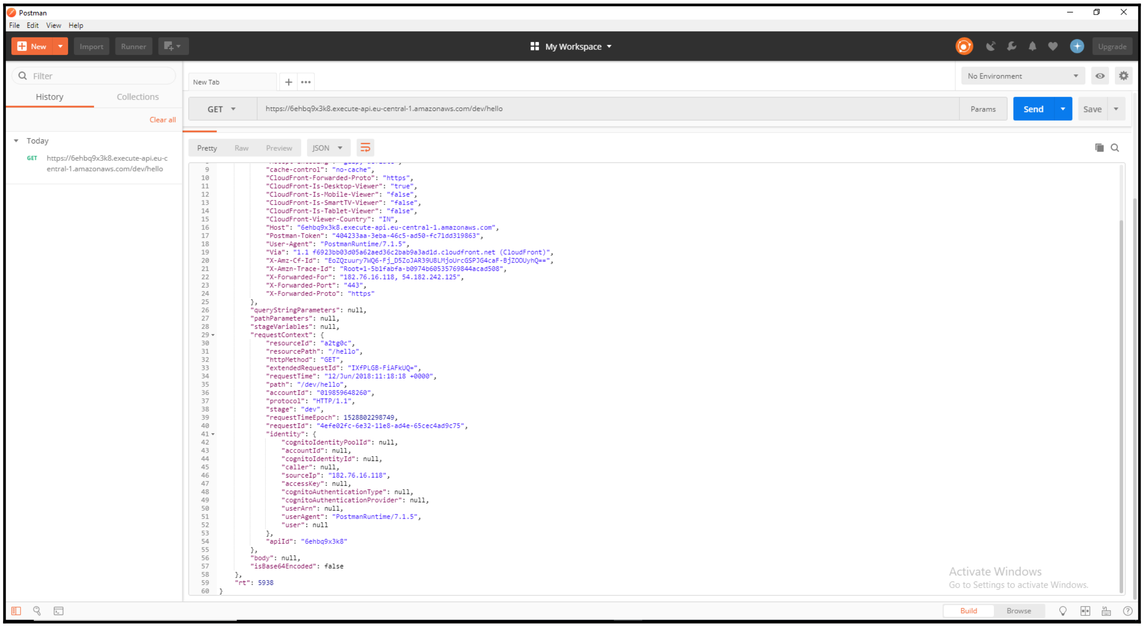
```
{
    "statusCode": 200,
    "ev": {
        "foo": "bar"
    },
    "rt": 6000,
    "fn": "simple-dev-hello",
    "aid": "id"
}
```

```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app> sls invoke local -f hello -p event.json
event is { foo: 'bar' }
context is { awsRequestId: 'id',
  invokeid: 'id',
  logGroupName: '/aws/lambda/simple-dev-hello',
  logStreamName: '2015/09/22/[HEAD]13370a84ca4ed8b77c427af260',
  functionVersion: 'HEAD',
  isDefaultFunctionVersion: true,
  functionName: 'simple-dev-hello',
  memoryLimitInMB: '1024',
  succeed: [Function: succeed],
  fail: [Function: fail],
  done: [Function: done],
  getRemainingTimeInMillis: [Function: getRemainingTimeInMillis] }
{
    "statusCode": 200,
    "body": "{\"ev\":{\"foo\":\"bar\"},\"rt\":6000}"
}
```

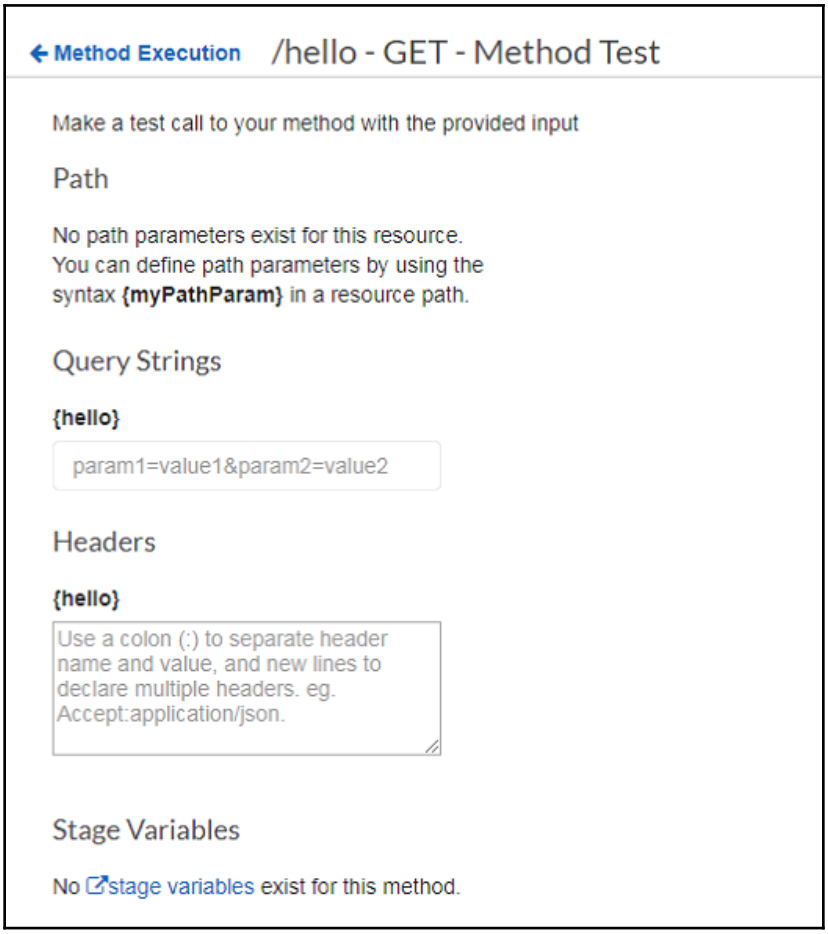
```
config ..................... Configure Serverless
config credentials ......... Configures a new provider profile for the Serverless Framework
create ..................... Create new Serverless service
deploy ..................... Deploy a Serverless service
deploy function ............ Deploy a single function from the service
deploy list ................ List deployed version of your Serverless Service
deploy list functions ...... List all the deployed functions and their versions
info ....................... Display information about the service
install .................... Install a Serverless service from GitHub or a plugin from the Serverless registry
invoke ..................... Invoke a deployed function
invoke local ............... Invoke function locally
logs ....................... Output the logs of a deployed function
metrics .................... Show metrics for a specific function
```

```
:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>sls logs -f hello
START RequestId: 2e2f66f0-6e1f-11e8-bc0f-2dedd2183214 Version: $LATEST
2018-06-12 14:31:23.540 (+05:30)        2e2f66f0-6e1f-11e8-bc0f-2dedd2183214    event is { foo: 'bar' }
2018-06-12 14:31:23.607 (+05:30)        2e2f66f0-6e1f-11e8-bc0f-2dedd2183214    context is { callbackWaitsForEmptyEventLoop: [Getter/Setter],
  done: [Function],
  succeed: [Function],
  fail: [Function],
  logGroupName: '/aws/lambda/simple-dev-hello',
  logStreamName: '2018/06/12/[$LATEST]3e6ab0888f1d4af9b310ff2b986e80d0',
  functionName: 'simple-dev-hello',
```

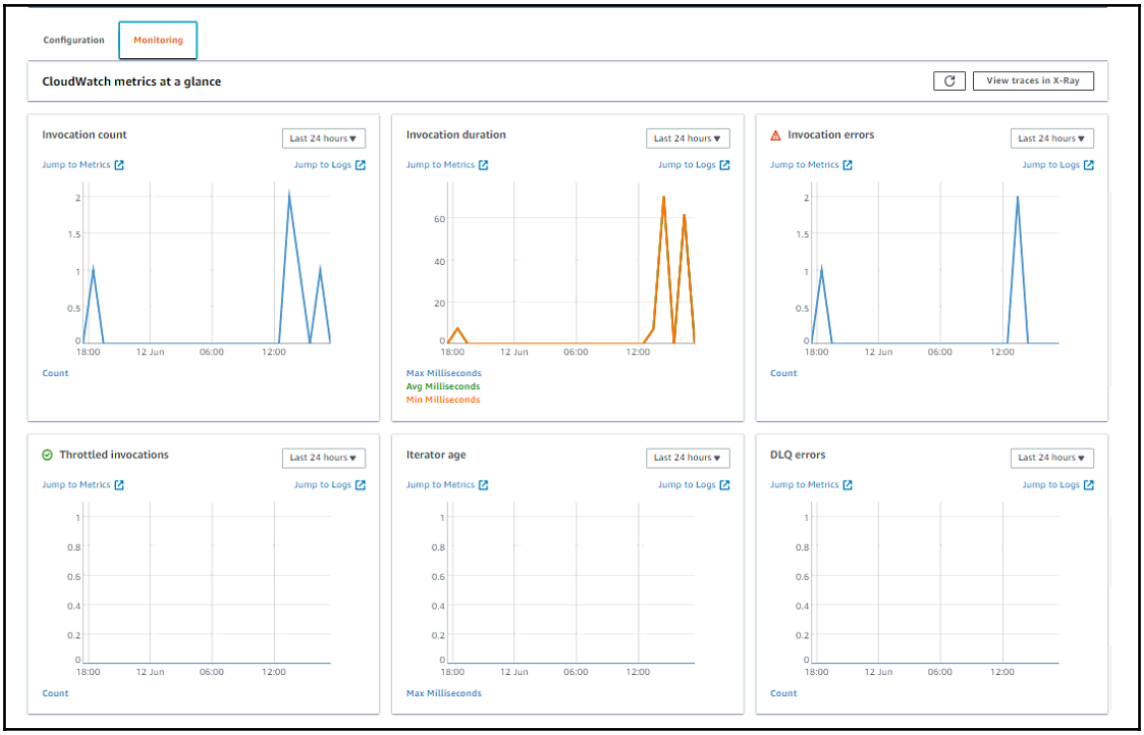
```
  memoryLimitInMB: '1024',
  functionVersion: '$LATEST',
  getRemainingTimeInMillis: [Function],
  invokeid: '2e2f66f0-6e1f-11e8-bc0f-2dedd2183214',
  awsRequestId: '2e2f66f0-6e1f-11e8-bc0f-2dedd2183214',
  invokedFunctionArn: 'arn:aws:lambda:eu-central-1:019859648260:function:simple-dev-hello' }
END RequestId: 2e2f66f0-6e1f-11e8-bc0f-2dedd2183214
REPORT RequestId: 2e2f66f0-6e1f-11e8-bc0f-2dedd2183214  Duration: 70.29 ms      Billed Duration: 100 ms         Memory Size: 1024 MB    Max Memory Used: 22 MB
```

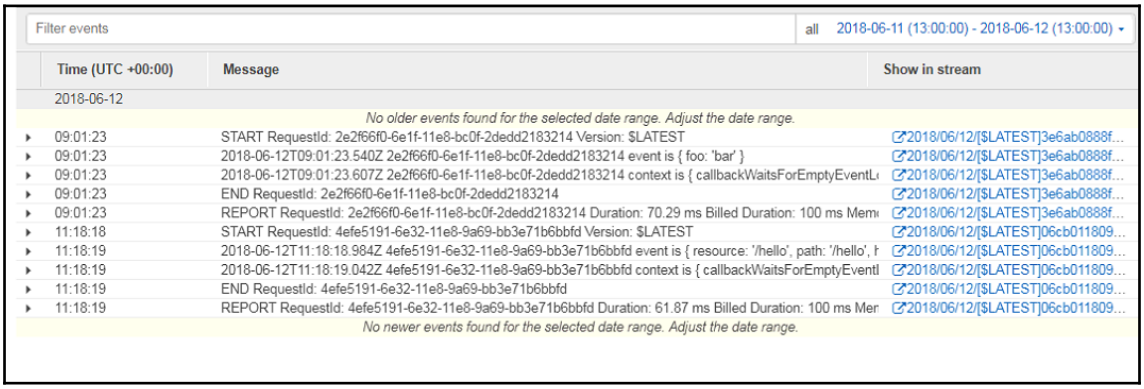


Postman
File Edit View Help
New
Import
Runner
My Workspace
Upgrade
Filter
History
Collections
Clear all
Today
GET https://6ehbq9x3k8.execute-api.eu-central-1.amazonaws.com/dev/hello
New Tab
No Environment
GET
https://6ehbq9x3k8.execute-api.eu-central-1.amazonaws.com/dev/hello
Params
Send
Save
Pretty
Raw
Preview
JSON
"cache-control": "no-cache",
"CloudFront-Forwarded-Proto": "https",
"CloudFront-Is-Desktop-Viewer": "true",
"CloudFront-Is-Mobile-Viewer": "false",
"CloudFront-Is-SmartTV-Viewer": "false",
"CloudFront-Is-Tablet-Viewer": "false",
"CloudFront-Viewer-Country": "IN",
"Host": "6ehbq9x3k8.execute-api.eu-central-1.amazonaws.com",
"Postman-Token": "404233aa-3eba-46c5-ad50-fc71dd319863",
"User-Agent": "PostmanRuntime/7.1.5",
"Via": "1.1 f6923bb03d05a62aed36c2bab9a3ad1d.cloudfront.net (CloudFront)",
"X-Amz-Cf-Id": "EoZQzuury7WQ6-Fj_D5ZoJAR39U8LMjoUrcGSPJG4caF-BjZOOUyhQ==",
"X-Amzn-Trace-Id": "Root=1-5b1fabfa-b0974b60535769844acad508",
"X-Forwarded-For": "182.76.16.118, 54.182.242.125",
"X-Forwarded-Port": "443",
"X-Forwarded-Proto": "https"
},
"queryStringParameters": null,
"pathParameters": null,
"stageVariables": null,
"requestContext": {
"resourceId": "a2tg0c",
"resourcePath": "/hello",
"httpMethod": "GET",
"extendedRequestId": "IXfPLGB-FiAFkUQ=",
"requestTime": "12/Jun/2018:11:18:18 +0000",
"path": "/dev/hello",
"accountId": "019859648260",
"protocol": "HTTP/1.1",
"stage": "dev",
"requestTimeEpoch": 1528802298749,
"requestId": "4efe02fc-6e32-11e8-ad4e-65cec4ad9c75",
"identity": {
"cognitoIdentityPoolId": null,
"accountId": null,
"cognitoIdentityId": null,
"caller": null,
"sourceIp": "182.76.16.118",
"accessKey": null,
"cognitoAuthenticationType": null,
"cognitoAuthenticationProvider": null,
"userArn": null,
"userAgent": "PostmanRuntime/7.1.5",
"user": null
},
"apiId": "6ehbq9x3k8"
},
"body": null,
"isBase64Encoded": false
},
"rt": 5938
}
Activate Windows
Go to Settings to activate Windows.
Build
Browse


Method Execution
/hello - GET - Method Test
Make a test call to your method with the provided input
Path
No path parameters exist for this resource.
You can define path parameters by using the
syntax {myPathParam} in a resource path.
Query Strings
{hello}
param1=value1&param2=value2
Headers
{hello}
Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. Accept:application/json.
Stage Variables
No stage variables exist for this method.


Lambda simple-dev-hello

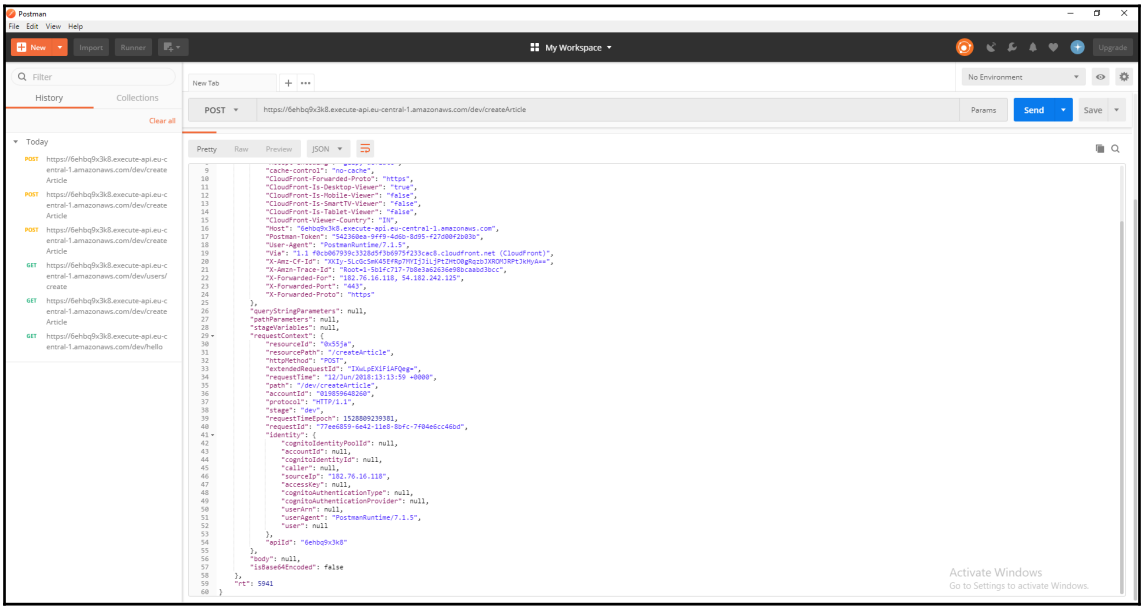

Configuration
Monitoring
CloudWatch metrics at a glance
View traces in X-Ray
Invocation count
Last 24 hours
Jump to Metrics
Jump to Logs
Count
Invocation duration
Max Milliseconds
Avg Milliseconds
Min Milliseconds
Invocation errors
Throttled invocations
Iterator age
DLQ errors
18:00
12 Jun
06:00
12:00


Filter events
all
2018-06-11 (13:00:00) - 2018-06-12 (13:00:00)
Time (UTC +00:00)
Message
Show in stream
2018-06-12
No older events found for the selected date range. Adjust the date range.
09:01:23 START RequestId: 2e2f66f0-6e1f-11e8-bc0f-2dedd2183214 Version: $LATEST
09:01:23 2018-06-12T09:01:23.540Z 2e2f66f0-6e1f-11e8-bc0f-2dedd2183214 event is { foo: 'bar' }
09:01:23 2018-06-12T09:01:23.607Z 2e2f66f0-6e1f-11e8-bc0f-2dedd2183214 context is { callbackWaitsForEmptyEventL
09:01:23 END RequestId: 2e2f66f0-6e1f-11e8-bc0f-2dedd2183214
09:01:23 REPORT RequestId: 2e2f66f0-6e1f-11e8-bc0f-2dedd2183214 Duration: 70.29 ms Billed Duration: 100 ms Mem
11:18:18 START RequestId: 4efe5191-6e32-11e8-9a69-bb3e71b6bbfd Version: $LATEST
11:18:19 2018-06-12T11:18:18.984Z 4efe5191-6e32-11e8-9a69-bb3e71b6bbfd event is { resource: '/hello', path: '/hello', h
11:18:19 2018-06-12T11:18:19.042Z 4efe5191-6e32-11e8-9a69-bb3e71b6bbfd context is { callbackWaitsForEmptyEventl
11:18:19 END RequestId: 4efe5191-6e32-11e8-9a69-bb3e71b6bbfd
11:18:19 REPORT RequestId: 4efe5191-6e32-11e8-9a69-bb3e71b6bbfd Duration: 61.87 ms Billed Duration: 100 ms Mer
No newer events found for the selected date range. Adjust the date range.

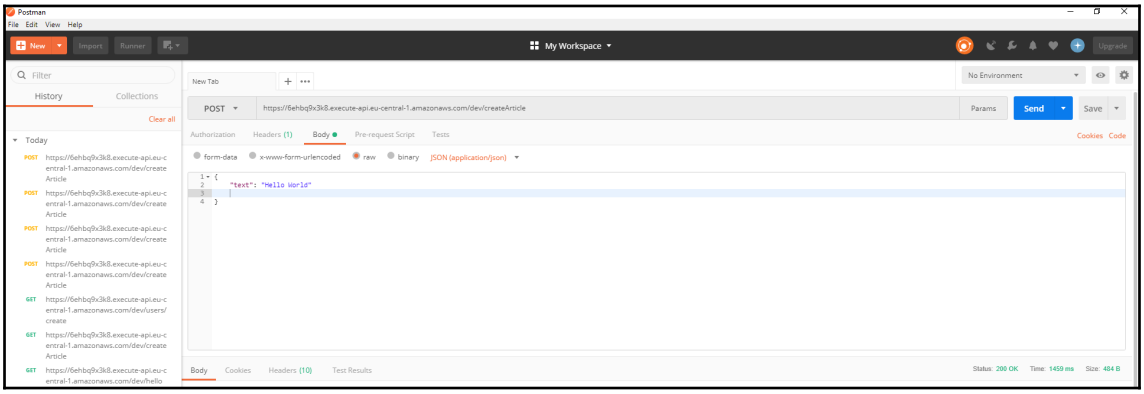
Chapter 3: Building a Serverless Application


```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>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 .zip file to S3 (5.96 KB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.....
Serverless: Stack update finished...
Service Information
service: blog
stage: dev
region: eu-central-1
stack: blog-dev
api keys:
  None
endpoints:
  POST - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  PUT - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/deleteArticle
functions:
  createArticle: blog-dev-createArticle
  readArticle: blog-dev-readArticle
  updateArticle: blog-dev-updateArticle
  deleteArticle: blog-dev-deleteArticle
```

```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>sls invoke -f createArticle
{
    "statusCode": 200,
    "body": "{\"ev\":{},\"rt\":5929}"
}
```




```
endpoints:
  POST - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  PUT - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/deleteArticle
```

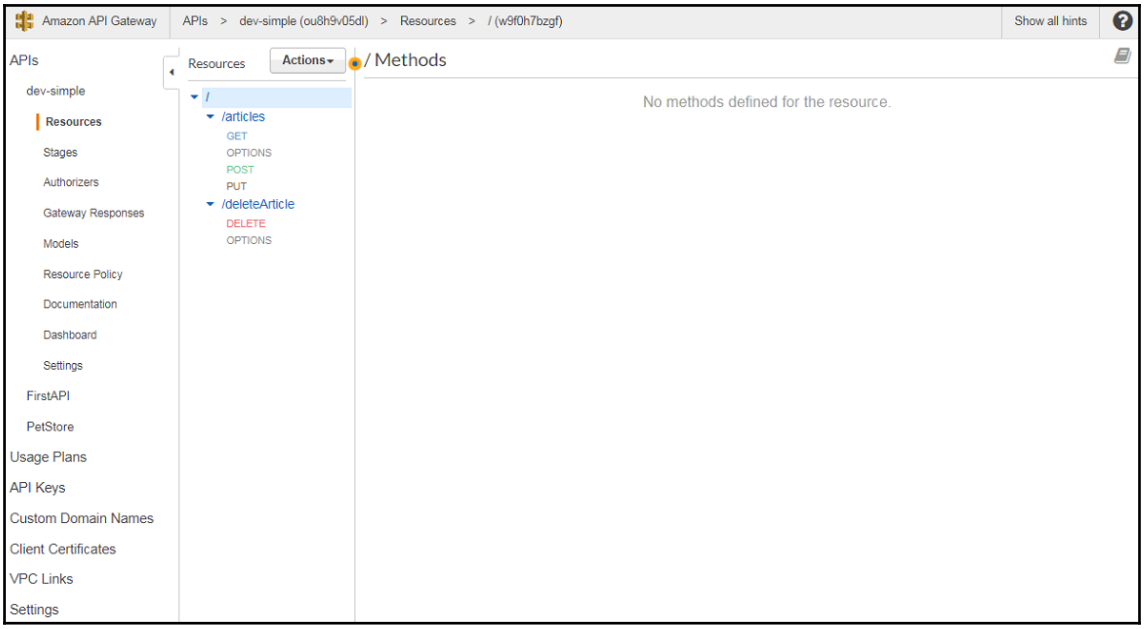


```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>sls logs -f createArticle1
START RequestId: ce694d85-6e40-11e8-9fed-1156a549baac Version: $LATEST
Unable to import module 'articles/create': Error
    at Function.Module._resolveFilename (module.js:325:15)
    at Function.Module._load (module.js:276:25)
    at Module.require (module.js:353:17)
    at require (internal/module.js:12:17)
    at Object.<anonymous> (/var/task/articles/model.js:3:14)
    at Module._compile (module.js:409:26)
    at Object.Module._extensions..js (module.js:416:10)
    at Module.load (module.js:343:32)
    at Function.Module._load (module.js:300:12)
    at Module.require (module.js:353:17)
END RequestId: ce694d85-6e40-11e8-9fed-1156a549baac
REPORT RequestId: ce694d85-6e40-11e8-9fed-1156a549baac  Duration: 6.92 ms       Billed Duration: 100 ms         Memory Size: 1024 MB    Max Memory Used: 39 MB

START RequestId: 315d0753-6e41-11e8-a156-f7a2f98ac002 Version: $LATEST
Unable to import module 'articles/create': Error
    at Function.Module._load (module.js:276:25)
    at Module.require (module.js:353:17)
    at require (internal/module.js:12:17)
    at Object.<anonymous> (/var/task/articles/model.js:3:14)
    at Module._compile (module.js:409:26)
    at Object.Module._extensions..js (module.js:416:10)
    at Module.load (module.js:343:32)
    at Function.Module._load (module.js:300:12)
    at Module.require (module.js:353:17)
END RequestId: 315d0753-6e41-11e8-a156-f7a2f98ac002
REPORT RequestId: 315d0753-6e41-11e8-a156-f7a2f98ac002  Duration: 2.30 ms       Billed Duration: 100 ms         Memory Size: 1024 MB    Max Memory Used: 39 MB

START RequestId: 21d7aec0-6e42-11e8-9f70-87598087df0b Version: $LATEST
Unable to import module 'articles/create': Error
    at Function.Module._resolveFilename (module.js:325:15)
    at Function.Module._load (module.js:276:25)
    at Module.require (module.js:353:17)
    at require (internal/module.js:12:17)
    at Object.<anonymous> (/var/task/articles/model.js:3:14)
    at Module._compile (module.js:409:26)
    at Object.Module._extensions..js (module.js:416:10)
    at Module.load (module.js:343:32)
    at Function.Module._load (module.js:300:12)
    at Module.require (module.js:353:17)
END RequestId: 21d7aec0-6e42-11e8-9f70-87598087df0b
REPORT RequestId: 21d7aec0-6e42-11e8-9f70-87598087df0b  Duration: 6.88 ms       Billed Duration: 100 ms         Memory Size: 1024 MB    Max Memory Used: 35 MB

START RequestId: ef29a59a-6e46-11e8-a990-8303bf9c4835 Version: $LATEST
2018-06-12 19:15:57.844 (+05:30)        ef29a59a-6e46-11e8-a990-8303bf9c4835    Hello World
END RequestId: ef29a59a-6e46-11e8-a990-8303bf9c4835
REPORT RequestId: ef29a59a-6e46-11e8-a990-8303bf9c4835  Duration: 3.22 ms       Billed Duration: 100 ms         Memory Size: 1024 MB    Max Memory Used: 21 MB
```

```
START RequestId: ef29a59a-6e46-11e8-a990-8303bf9c4835 Version: $LATEST
2018-06-12 19:15:57.844 (+05:30)        ef29a59a-6e46-11e8-a990-8303bf9c4835    Hello World
END RequestId: ef29a59a-6e46-11e8-a990-8303bf9c4835
REPORT RequestId: ef29a59a-6e46-11e8-a990-8303bf9c4835  Duration: 3.22 ms       Billed Duration: 100 ms         Memory Size: 1024 MB    Max Memory Used: 21 MB
```

```
functions:
  createArticle: blog-dev-createArticle
  readArticle: blog-dev-readArticle
  updateArticle: blog-dev-updateArticle
  deleteArticle: blog-dev-deleteArticle
```

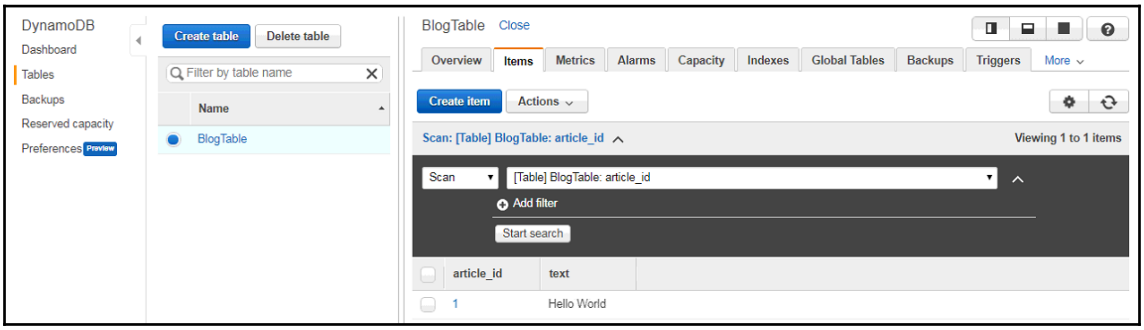
```
endpoints:
  POST - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  PUT - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/deleteArticle
```

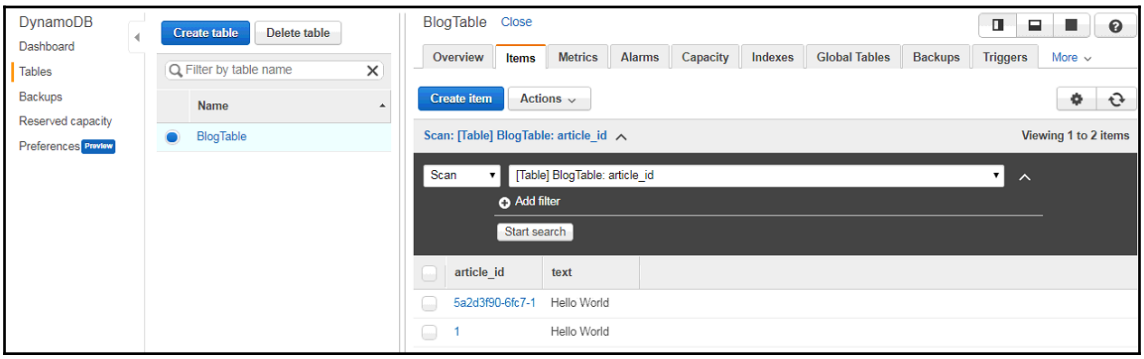



```
endpoints:
  POST - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  PUT - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://9owl38dug8.execute-api.eu-central-1.amazonaws.com/dev/deleteArticle
functions:
  createArticle: blog-dev-createArticle
  readArticle: blog-dev-readArticle
  updateArticle: blog-dev-updateArticle
  deleteArticle: blog-dev-deleteArticle
```

```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>npm init -y
Wrote to C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app\package.json:

{
  "name": "blog-app",
  "version": "1.0.0",
  "description": "",
  "main": "handler.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

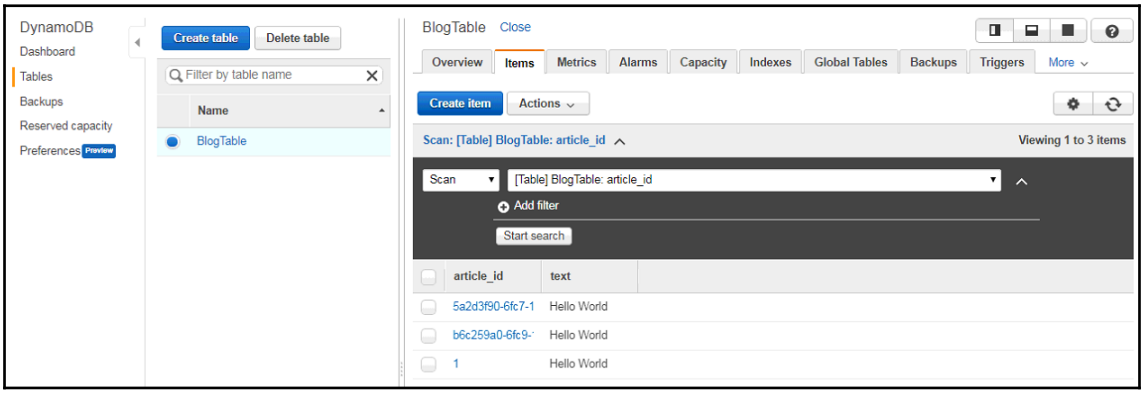
```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>sls invoke local -f createArticle -p articles/event.json
{
    "statusCode": 200
}

C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>
```

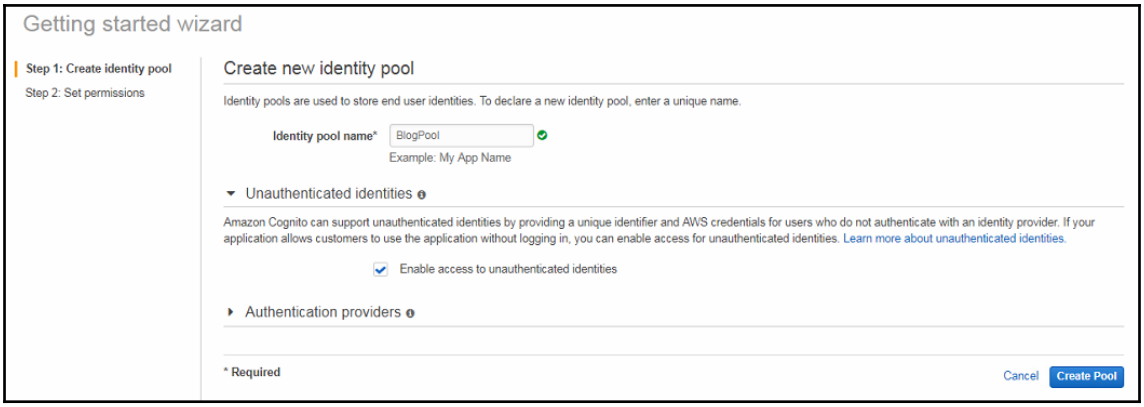


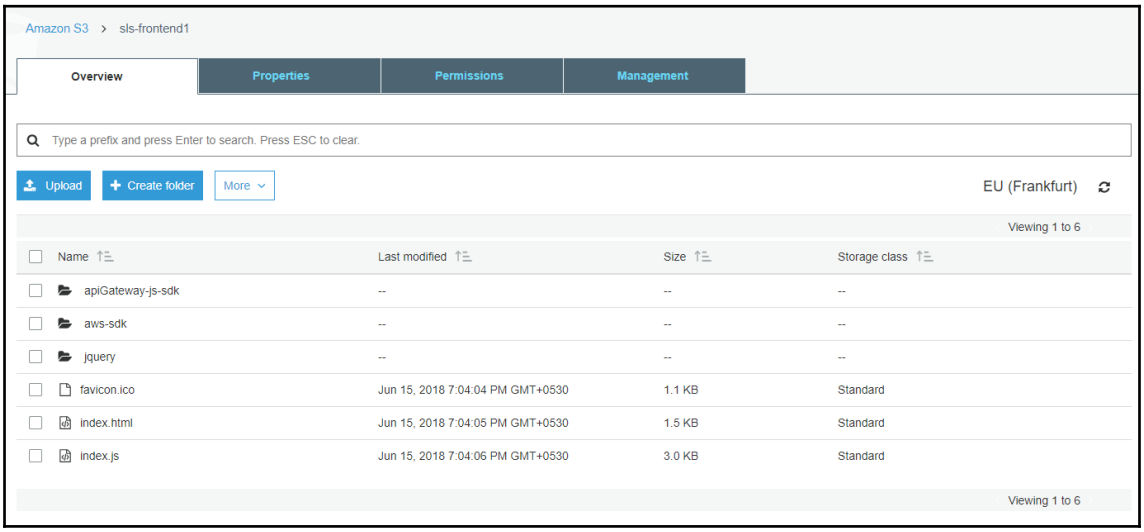


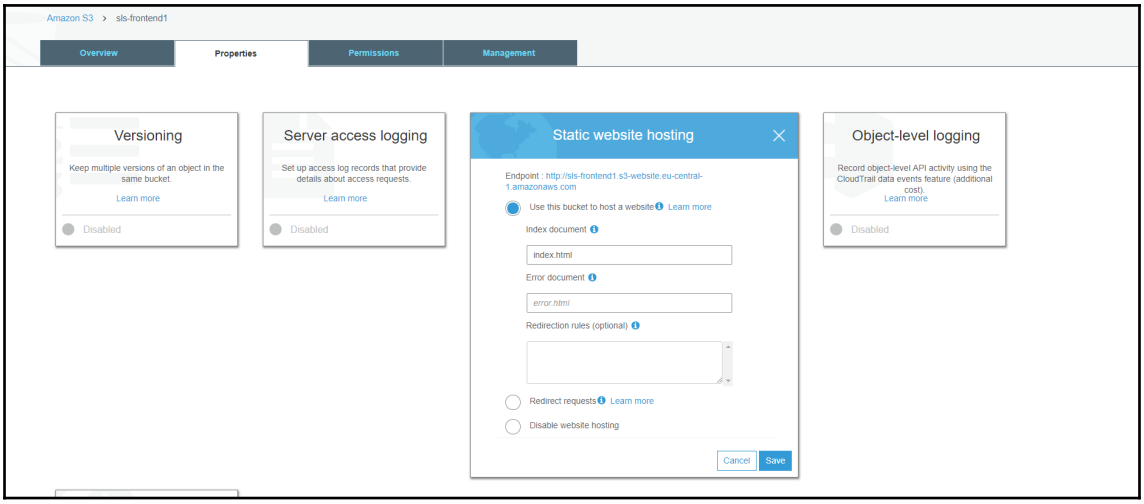

```
START RequestId: ab2a96ea-6fc7-11e8-bbb9-c79d86ee7e9f Version: $LATEST
2018-06-14 17:10:00.355 (+05:30)        ab2a96ea-6fc7-11e8-bbb9-c79d86ee7e9f    { [AccessDeniedException: User: arn:aws:sts::019859648260:assumed-role/blog-dev-eu-central-1-lambdaRole/blog-dev-createArticle is not authorized to perform:
dynamodb:PutItem on resource: arn:aws:dynamodb:eu-central-1:019859648260:table/BlogTable]
  message: 'User: arn:aws:sts::019859648260:assumed-role/blog-dev-eu-central-1-lambdaRole/blog-dev-createArticle is not authorized to perform: dynamodb:PutItem on resource: arn:aws:dynamodb:eu-central-1:019859648260:table/BlogTable',
  code: 'AccessDeniedException',
  time: Thu Jun 14 2018 11:40:00 GMT+0000 (UTC),
  requestId: '9A8J73OB69EA3LV165SH17V2LRVV4KQNSOSAEMVJF66Q9ASUAAJG',
  statusCode: 400,
  retryable: false,
  retryDelay: 10.440010103024543 }
```

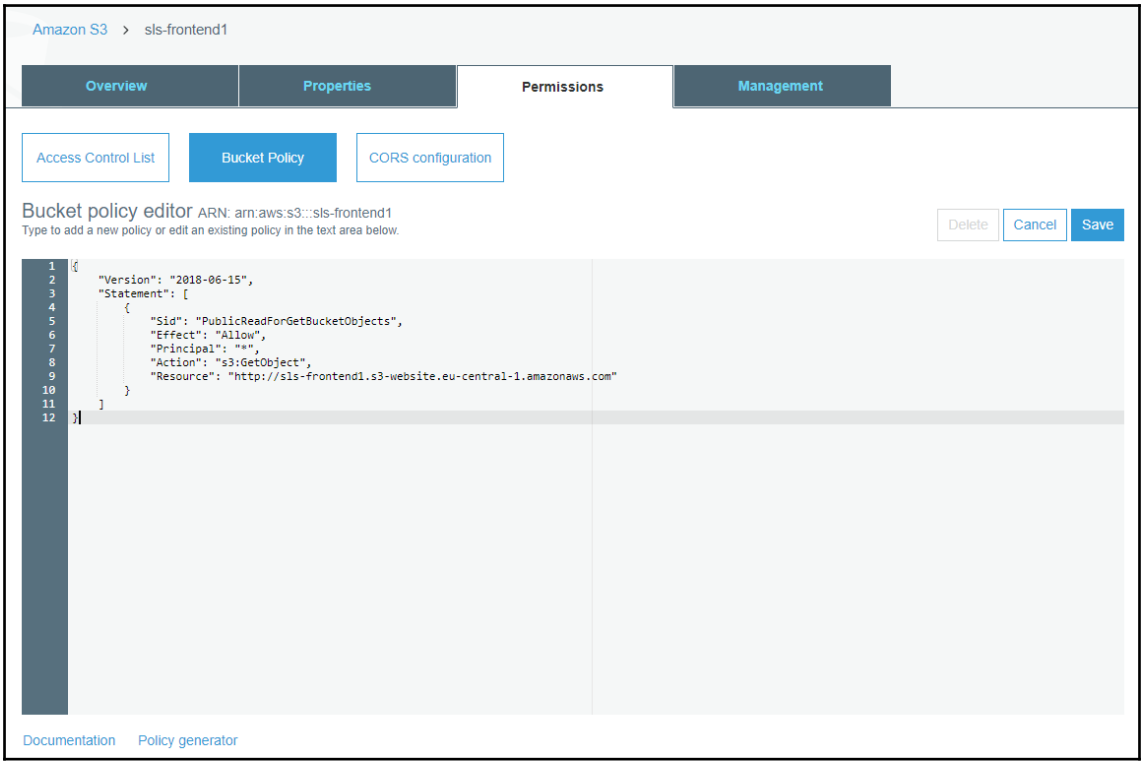
```
C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>sls deploy
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service .zip file to S3 (6.48 MB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.................
Serverless: Stack update finished...
Service Information
service: blog
stage: dev
region: eu-central-1
stack: blog-dev
api keys:
  None
endpoints:
  POST - https://jdonclv412.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://jdonclv412.execute-api.eu-central-1.amazonaws.com/dev/articles
  PUT - https://jdonclv412.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://jdonclv412.execute-api.eu-central-1.amazonaws.com/dev/deleteArticle
functions:
  createArticle: blog-dev-createArticle
  readArticle: blog-dev-readArticle
  updateArticle: blog-dev-updateArticle
  deleteArticle: blog-dev-deleteArticle

C:\Users\admin\Desktop\programming-aws-lambda-master\javascript\blog-app>sls invoke -f createArticle -p articles/event.json
{
    "statusCode": 200
}
```



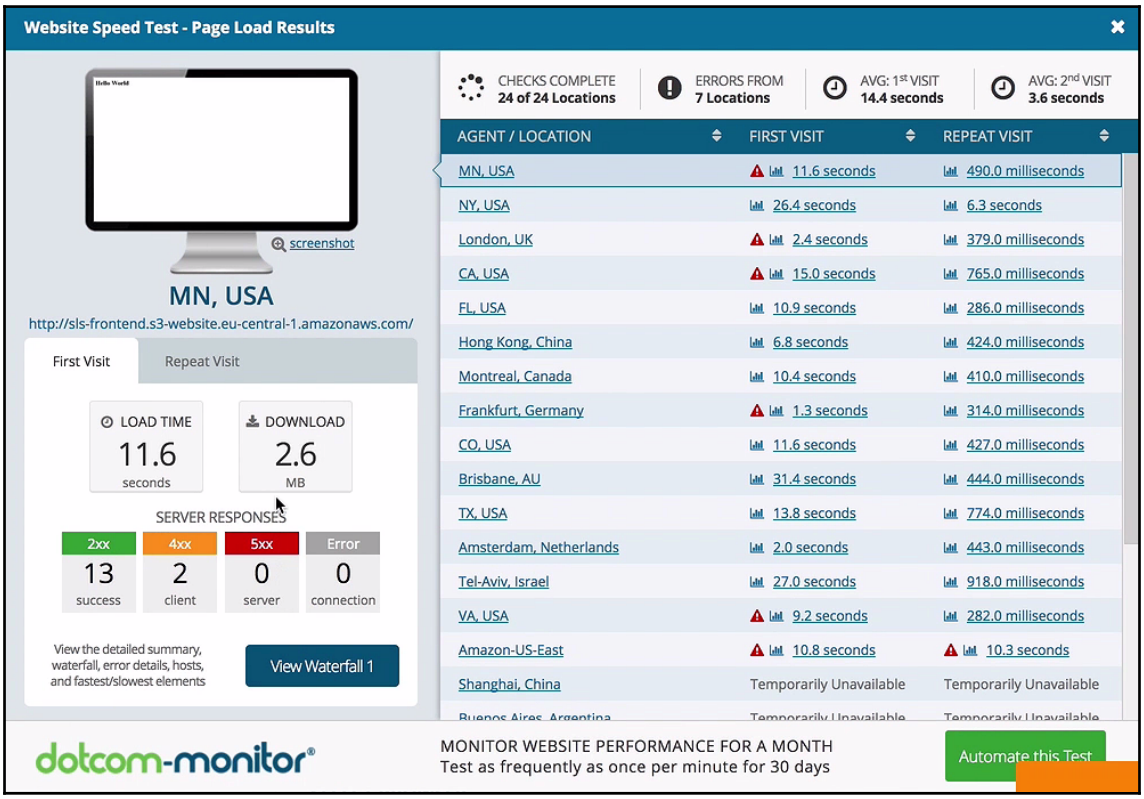

Getting started wizard
Step 1: Create identity pool
Step 2: Set permissions
Create new identity pool
Identity pools are used to store end user identities. To declare a new identity pool, enter a unique name.
Identity pool name*
BlogPool
Example: My App Name
Unauthenticated identities
Amazon Cognito can 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

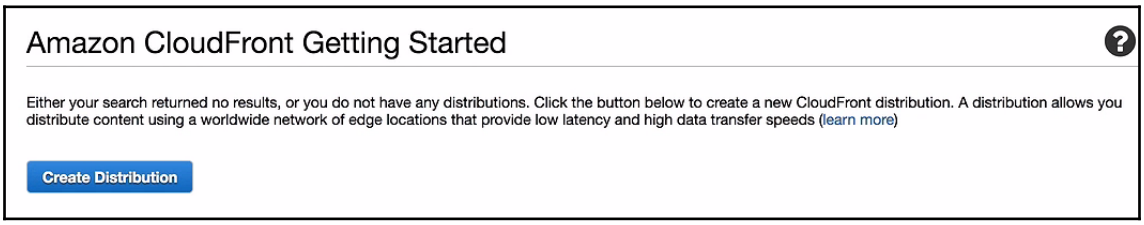

Amazon S3 > sls-frontend1
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
EU (Frankfurt)
Viewing 1 to 6
Name
Last modified
Size
Storage class
apiGateway-js-sdk -- -- --
aws-sdk -- -- --
jquery -- -- --
favicon.ico Jun 15, 2018 7:04:04 PM GMT+0530 1.1 KB Standard
index.html Jun 15, 2018 7:04:05 PM GMT+0530 1.5 KB Standard
index.js Jun 15, 2018 7:04:06 PM GMT+0530 3.0 KB Standard
Viewing 1 to 6


Amazon S3 › sls-frontend1
Overview
Properties
Permissions
Management
Versioning
Keep multiple versions of an object in the same bucket.
Learn more
Disabled
Server access logging
Set up access log records that provide details about access requests.
Learn more
Disabled
Static website hosting
Endpoint : http://sls-frontend1.s3-website.eu-central-1.amazonaws.com
Use this bucket to host a website Learn more
Index document
index.html
Error document
error.html
Redirection rules (optional)
Redirect requests Learn more
Disable website hosting
Cancel
Save
Object-level logging
Record object-level API activity using the CloudTrail data events feature (additional cost).
Learn more
Disabled

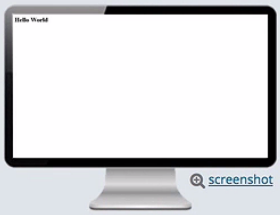

Amazon S3 › sls-frontend1
Overview
Properties
Permissions
Management
Access Control List
Bucket Policy
CORS configuration
Bucket policy editor ARN: arn:aws:s3:::sls-frontend1
Type to add a new policy or edit an existing policy in the text area below.
Delete
Cancel
Save
{
"Version": "2018-06-15",
"Statement": [
{
"Sid": "PublicReadForGetBucketObjects",
"Effect": "Allow",
"Principal": "*",
"Action": "s3:GetObject",
"Resource": "http://sls-frontend1.s3-website.eu-central-1.amazonaws.com"
}
]
}
Documentation
Policy generator

Hello World


Website Speed Test - Page Load Results
screenshot
MN, USA
http://sls-frontend.s3-website.eu-central-1.amazonaws.com/
First Visit
Repeat Visit
LOAD TIME
11.6
seconds
DOWNLOAD
2.6
MB
SERVER RESPONSES
2xx
13
success
4xx
2
client
5xx
0
server
Error
0
connection
View the detailed summary, waterfall, error details, hosts, and fastest/slowest elements
View Waterfall 1
CHECKS COMPLETE
24 of 24 Locations
ERRORS FROM
7 Locations
AVG: 1st VISIT
14.4 seconds
AVG: 2nd VISIT
3.6 seconds
AGENT / LOCATION
FIRST VISIT
REPEAT VISIT
MN, USA 11.6 seconds 490.0 milliseconds
NY, USA 26.4 seconds 6.3 seconds
London, UK 2.4 seconds 379.0 milliseconds
CA, USA 15.0 seconds 765.0 milliseconds
FL, USA 10.9 seconds 286.0 milliseconds
Hong Kong, China 6.8 seconds 424.0 milliseconds
Montreal, Canada 10.4 seconds 410.0 milliseconds
Frankfurt, Germany 1.3 seconds 314.0 milliseconds
CO, USA 11.6 seconds 427.0 milliseconds
Brisbane, AU 31.4 seconds 444.0 milliseconds
TX, USA 13.8 seconds 774.0 milliseconds
Amsterdam, Netherlands 2.0 seconds 443.0 milliseconds
Tel-Aviv, Israel 27.0 seconds 918.0 milliseconds
VA, USA 9.2 seconds 282.0 milliseconds
Amazon-US-East 10.8 seconds 10.3 seconds
Shanghai, China Temporarily Unavailable Temporarily Unavailable
dotcom-monitor
MONITOR WEBSITE PERFORMANCE FOR A MONTH
Test as frequently as once per minute for 30 days
Automate this Test


Amazon CloudFront Getting Started
Either your search returned no results, or you do not have any distributions. Click the button below to create a new CloudFront distribution. A distribution allows you distribute content using a worldwide network of edge locations that provide low latency and high data transfer speeds (learn more)
Create Distribution

Website Speed Test - Page Load Results



screenshot

MN, USA

drcq92o7rz6m6.cloudfront.net

First Visit | Repeat Visit

LOAD TIME	DOWNLOAD
4.4 seconds	2.6 MB

SERVER RESPONSES

2xx	4xx	5xx	Error
13 success	0 client	0 server	0 connection

View the detailed summary, waterfall, error details, hosts, and fastest/slowest elements

View Waterfall 1

CHECKS COMPLETE **24 of 24 Locations** | ERRORS FROM **5 Locations** | AVG: 1st VISIT **9.5 seconds** | AVG: 2nd VISIT **2.1 seconds**

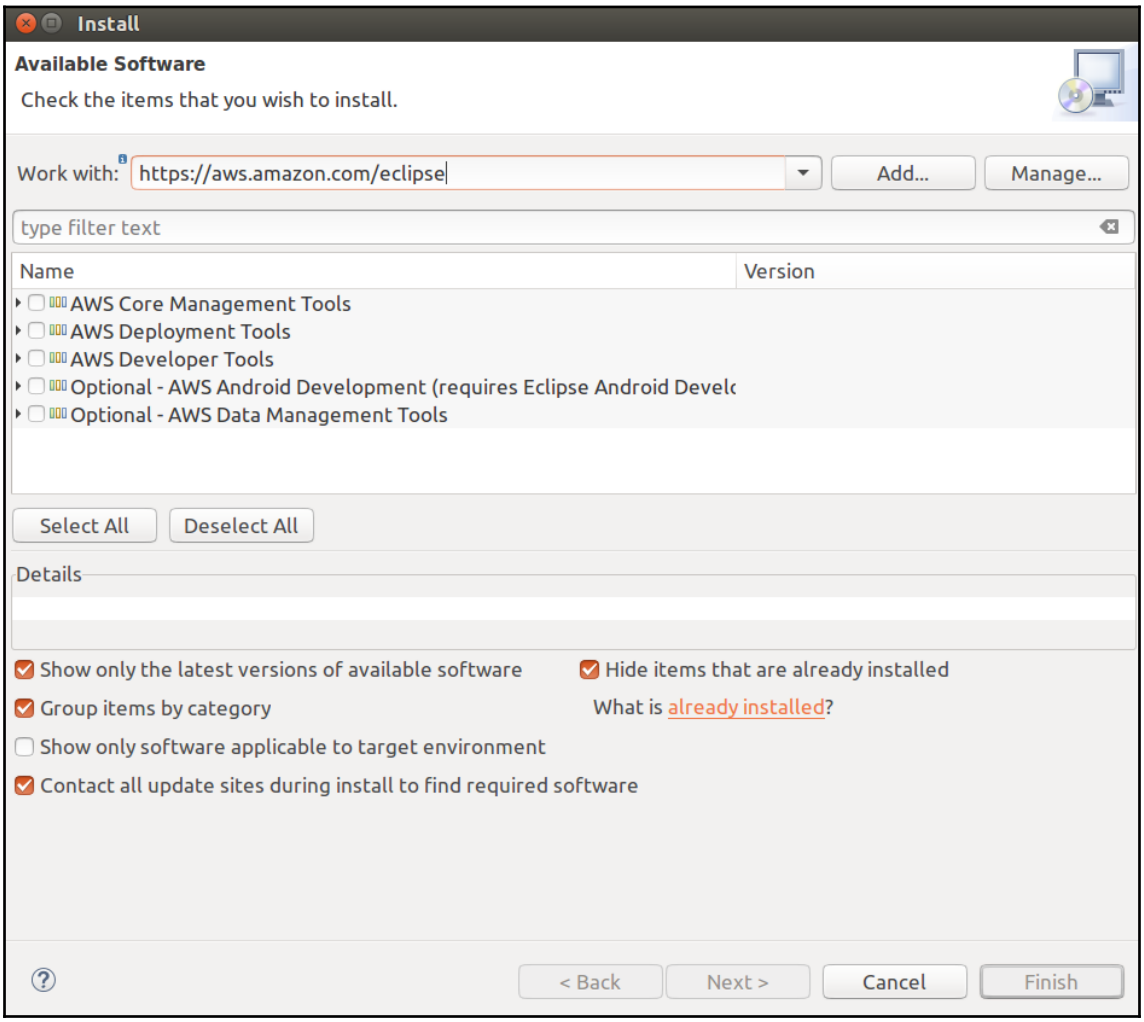
AGENT / LOCATION	FIRST VISIT	REPEAT VISIT
MN, USA	4.4 seconds	461.0 milliseconds
NY, USA	4.3 seconds	546.0 milliseconds
London, UK	1.2 seconds	318.0 milliseconds
CA, USA	6.0 seconds	317.0 milliseconds
FL, USA	5.0 seconds	283.0 milliseconds
Hong Kong, China	844.0 milliseconds	156.0 milliseconds
Montreal, Canada	3.7 seconds	467.0 milliseconds
Frankfurt, Germany	1.2 seconds	326.0 milliseconds
CO, USA	3.7 seconds	435.0 milliseconds
Brisbane, AU	9.8 seconds	455.0 milliseconds
TX, USA	4.1 seconds	404.0 milliseconds
Amsterdam, Netherlands	977.0 milliseconds	154.0 milliseconds
Tel-Aviv, Israel	25.3 seconds	645.0 milliseconds
VA, USA	3.2 seconds	277.0 milliseconds
Amazon-US-East	10.7 seconds	10.3 seconds
Shanghai, China	39.8 seconds	398.0 milliseconds
Buenos Aires, Argentina	10.9 seconds	11.3 seconds

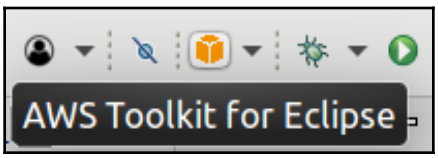
dotcom-monitor®

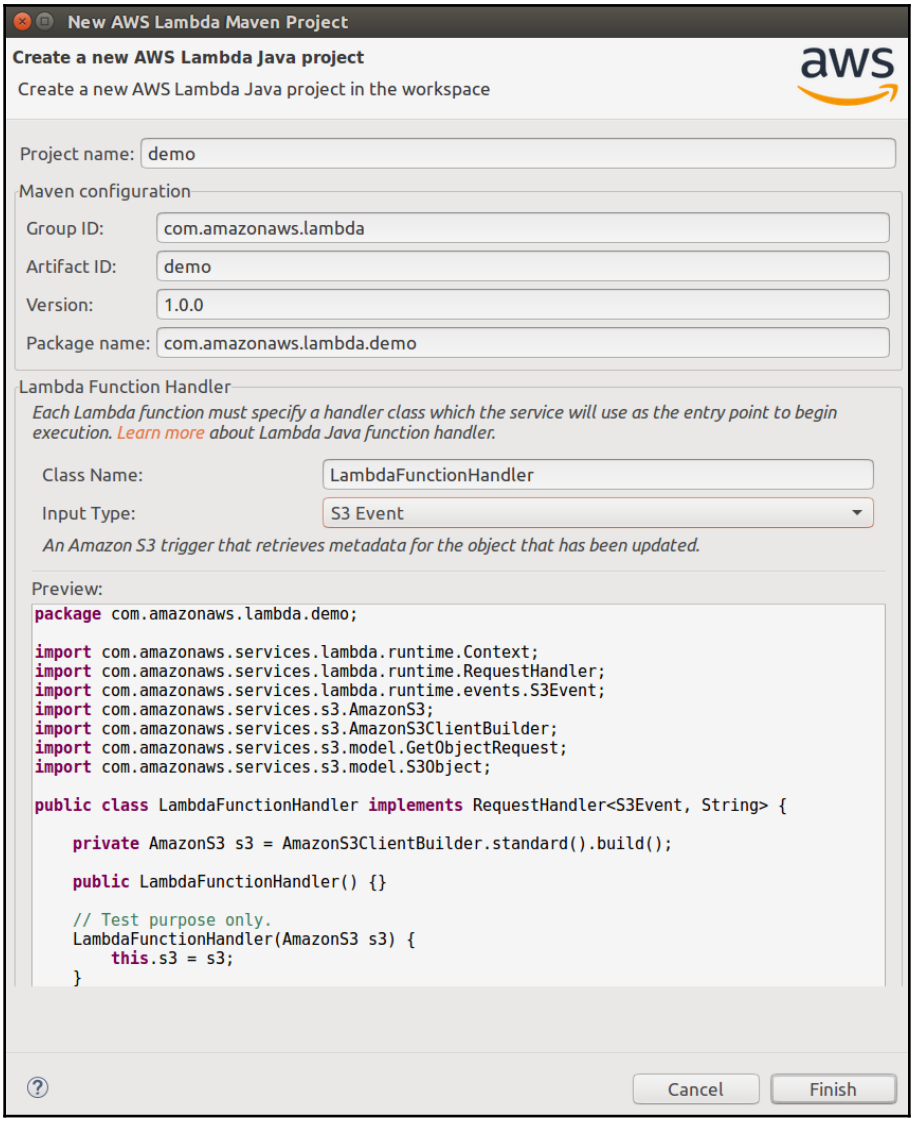
MONITOR WEBSITE PERFORMANCE FOR A MONTH
Test as frequently as once per minute for 30 days

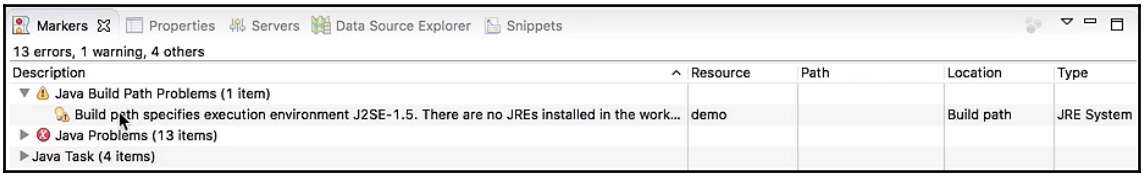
Automate this Test

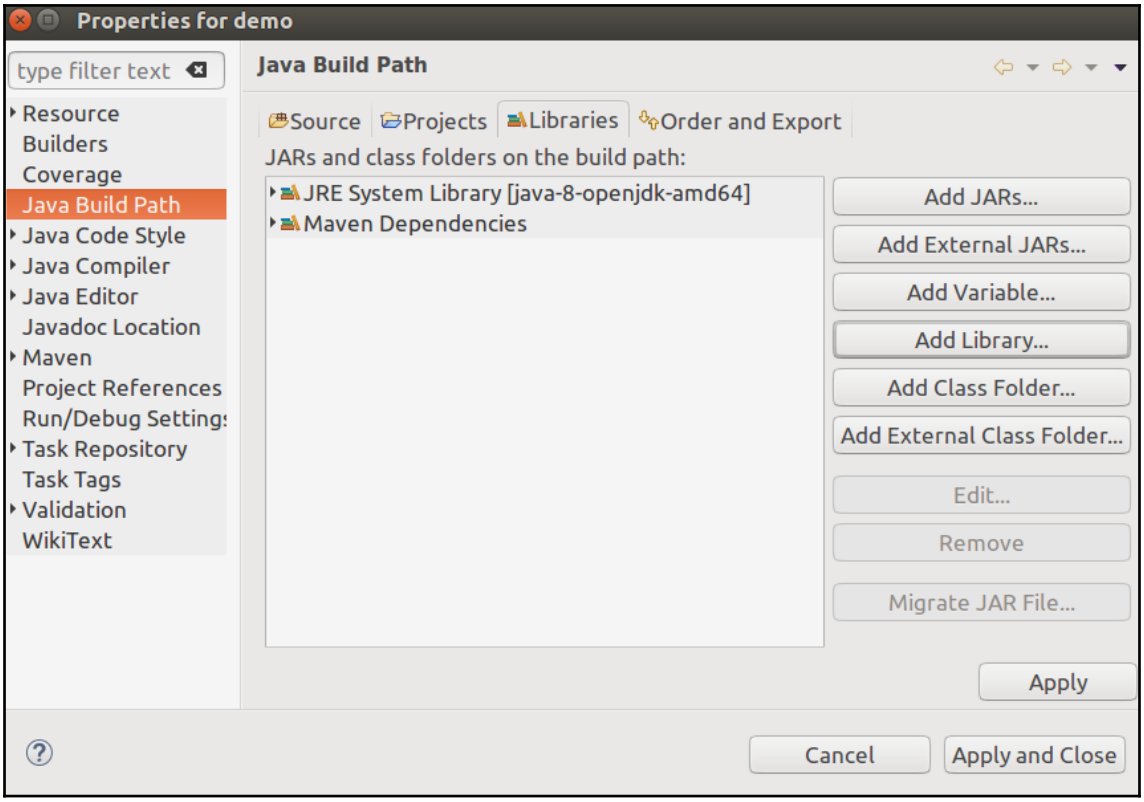
Chapter 4:
Programming AWS Lambda with Java

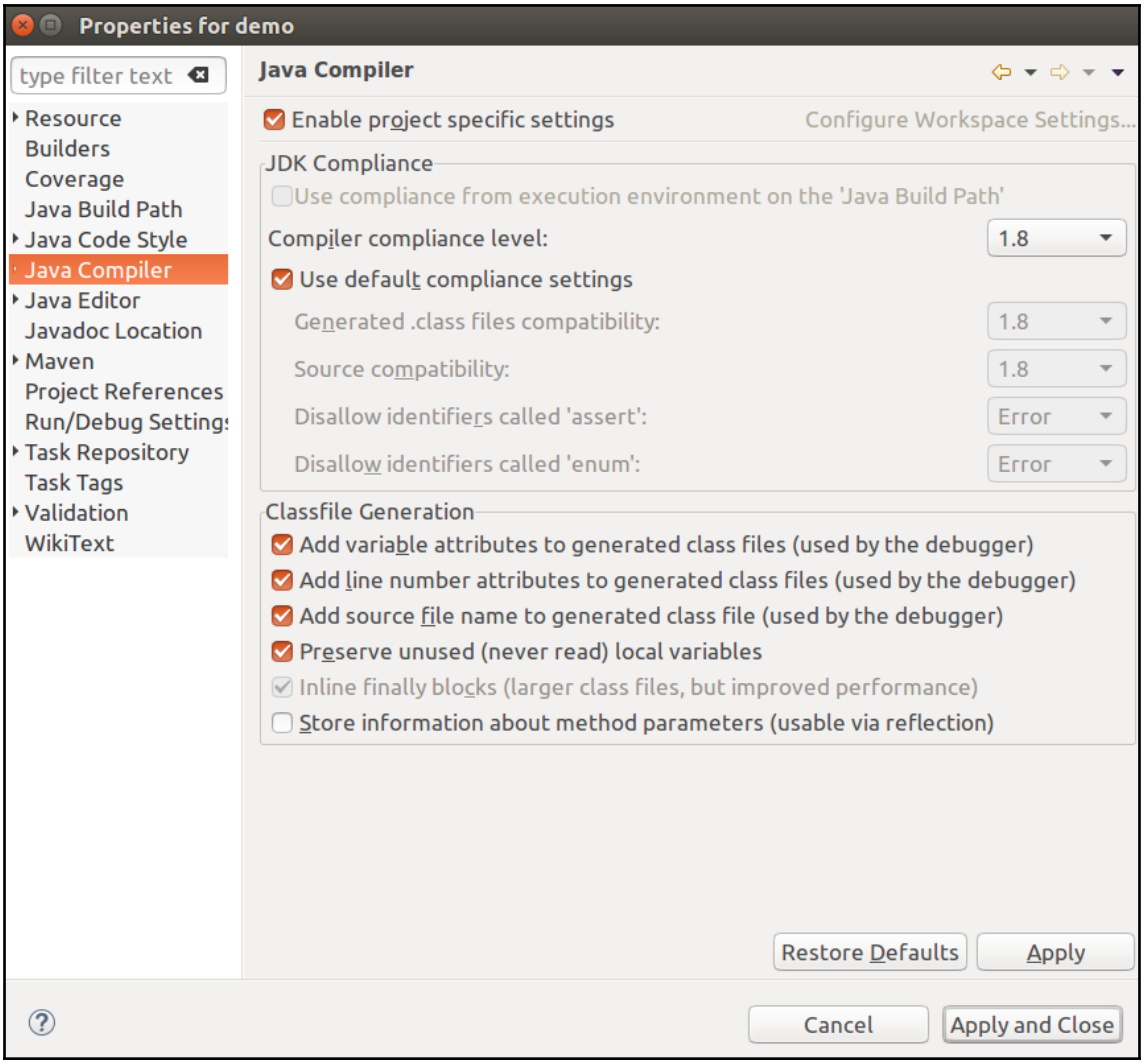


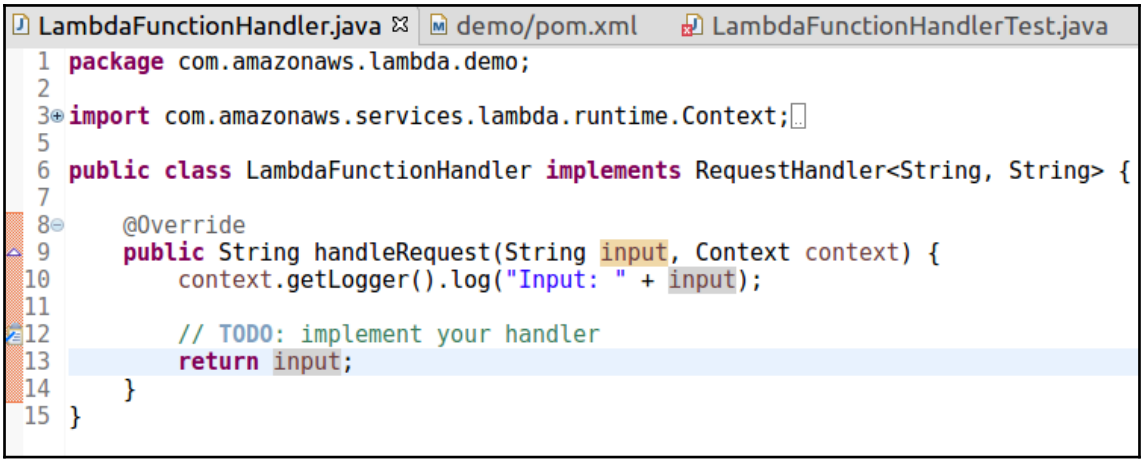


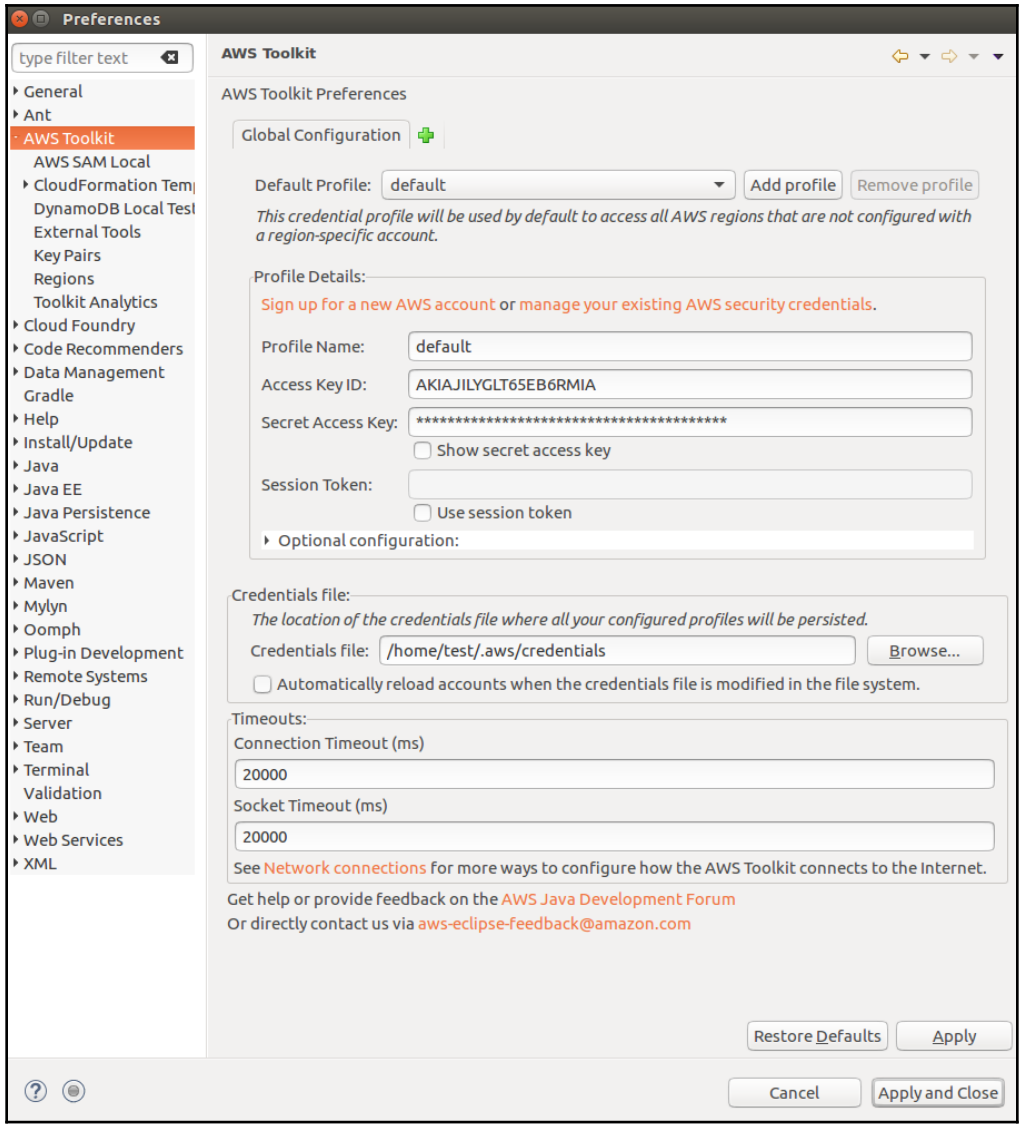

New AWS Lambda Maven Project
Create a new AWS Lambda Java project
Create a new AWS Lambda Java project in the workspace
aws
Project name: demo
Maven configuration
Group ID: com.amazonaws.lambda
Artifact ID: demo
Version: 1.0.0
Package name: com.amazonaws.lambda.demo
Lambda Function Handler
Each Lambda function must specify a handler class which the service will use as the entry point to begin execution. Learn more about Lambda Java function handler.
Class Name: LambdaFunctionHandler
Input Type: S3 Event
An Amazon S3 trigger that retrieves metadata for the object that has been updated.
Preview:
package com.amazonaws.lambda.demo;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.S3Event;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.S3Object;
public class LambdaFunctionHandler implements RequestHandler<S3Event, String> {
private AmazonS3 s3 = AmazonS3ClientBuilder.standard().build();
public LambdaFunctionHandler() {}
// Test purpose only.
LambdaFunctionHandler(AmazonS3 s3) {
this.s3 = s3;
}
Cancel
Finish

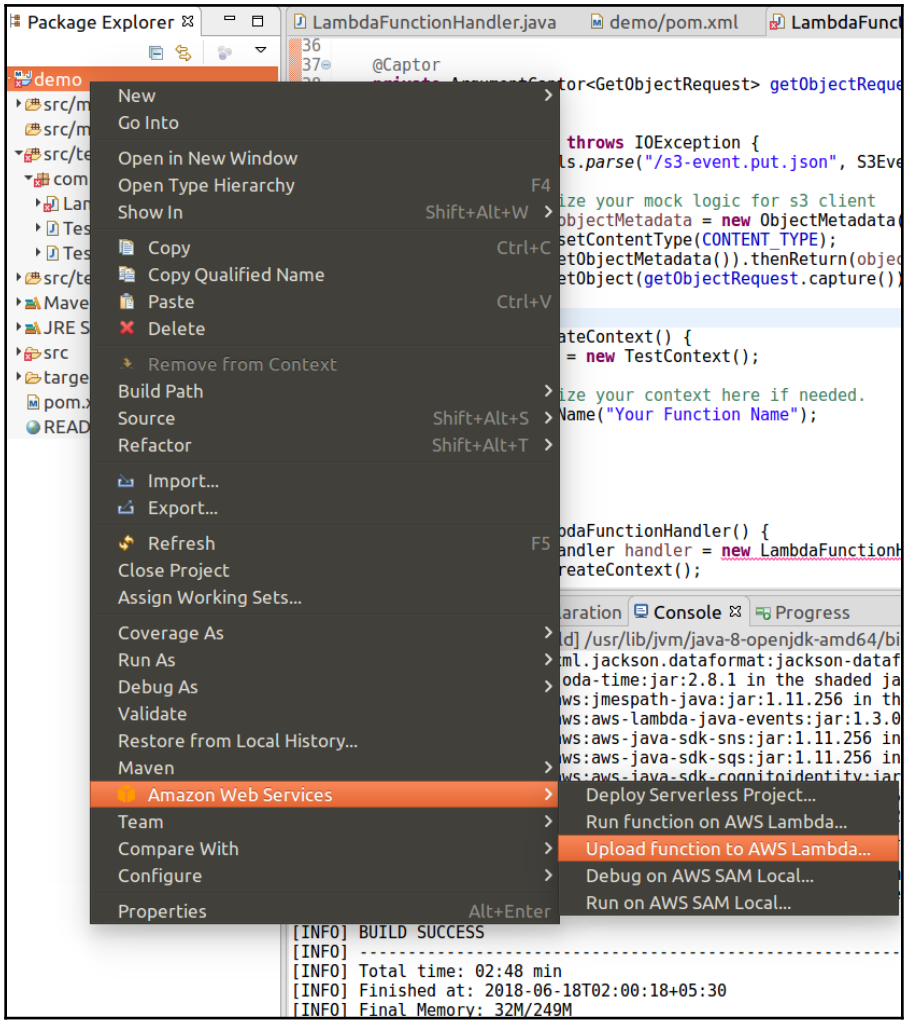

Markers
Properties
Servers
Data Source Explorer
Snippets
13 errors, 1 warning, 4 others
Description
Resource
Path
Location
Type
Java Build Path Problems (1 item)
Build path specifies execution environment J2SE-1.5. There are no JREs installed in the work...
demo
Build path
JRE System
Java Problems (13 items)
Java Task (4 items)

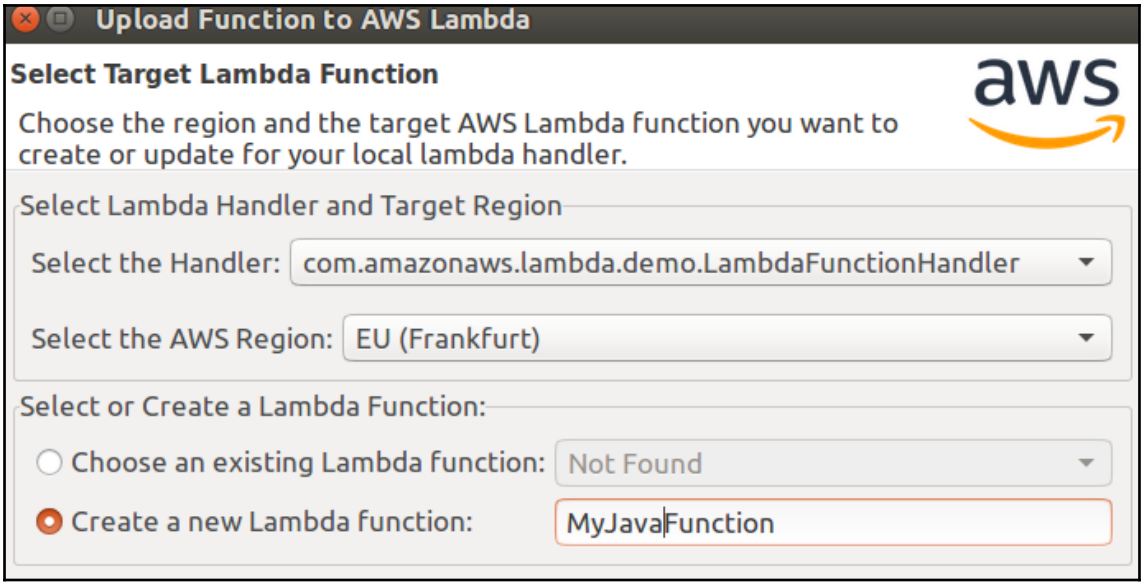

Properties for demo
type filter text
Resource
Builders
Coverage
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
Maven
Project References
Run/Debug Setting
Task Repository
Task Tags
Validation
WikiText
Java Build Path
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
JRE System Library [java-8-openjdk-amd64]
Maven Dependencies
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
Apply
Cancel
Apply and Close

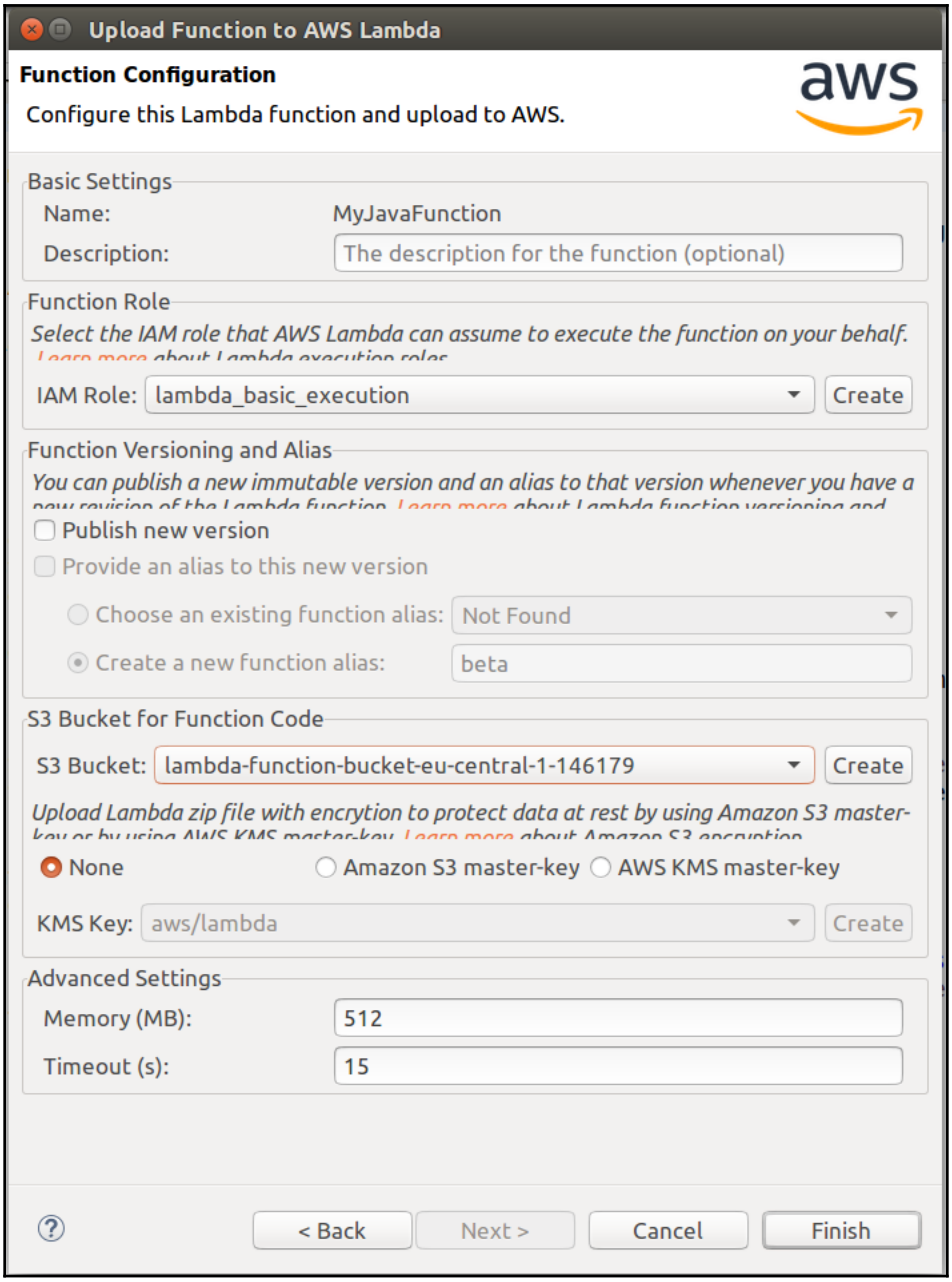

Properties for demo
type filter text
Resource
Builders
Coverage
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
Maven
Project References
Run/Debug Settings
Task Repository
Task Tags
Validation
WikiText
Java Compiler
Enable project specific settings
Configure Workspace Settings...
JDK Compliance
Use compliance from execution environment on the 'Java Build Path'
Compiler compliance level:
1.8
Use default compliance settings
Generated .class files compatibility:
1.8
Source compatibility:
1.8
Disallow identifiers called 'assert':
Error
Disallow identifiers called 'enum':
Error
Classfile Generation
Add variable attributes to generated class files (used by the debugger)
Add line number attributes to generated class files (used by the debugger)
Add source file name to generated class file (used by the debugger)
Preserve unused (never read) local variables
Inline finally blocks (larger class files, but improved performance)
Store information about method parameters (usable via reflection)
Restore Defaults
Apply
Cancel
Apply and Close

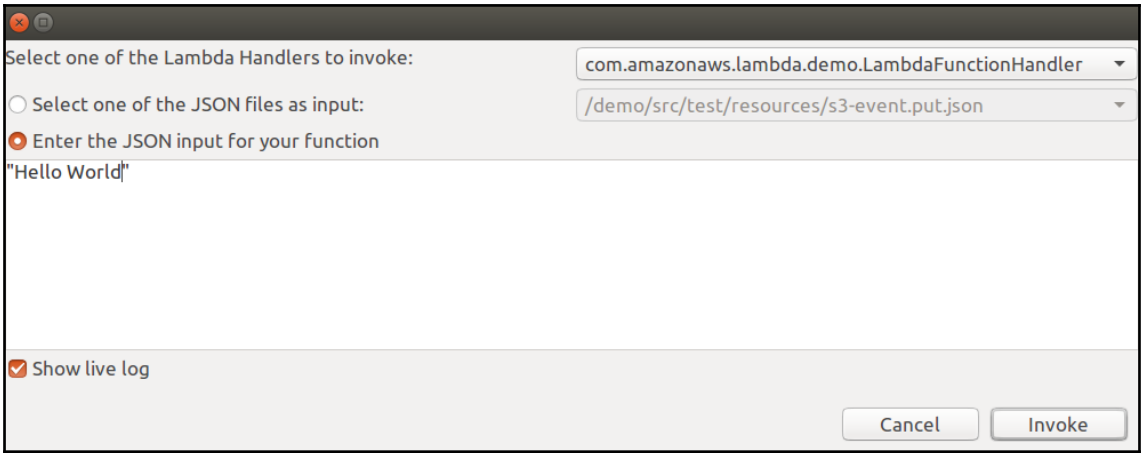

LambdaFunctionHandler.java
demo/pom.xml
LambdaFunctionHandlerTest.java
1 package com.amazonaws.lambda.demo;
2
3 import com.amazonaws.services.lambda.runtime.Context;
5
6 public class LambdaFunctionHandler implements RequestHandler<String, String> {
7
8 @Override
9 public String handleRequest(String input, Context context) {
10 context.getLogger().log("Input: " + input);
11
12 // TODO: implement your handler
13 return input;
14 }
15 }


Preferences
type filter text
General
Ant
AWS Toolkit
AWS SAM Local
CloudFormation Tem
DynamoDB Local Test
External Tools
Key Pairs
Regions
Toolkit Analytics
Cloud Foundry
Code Recommenders
Data Management
Gradle
Help
Install/Update
Java
Java EE
Java Persistence
JavaScript
JSON
Maven
Mylyn
Oomph
Plug-in Development
Remote Systems
Run/Debug
Server
Team
Terminal
Validation
Web
Web Services
XML
AWS Toolkit
AWS Toolkit Preferences
Global Configuration
Default Profile: default
Add profile
Remove profile
This credential profile will be used by default to access all AWS regions that are not configured with a region-specific account.
Profile Details:
Sign up for a new AWS account or manage your existing AWS security credentials.
Profile Name: default
Access Key ID: AKIAJILYGLT65EB6RMIA
Secret Access Key: ••
Show secret access key
Session Token:
Use session token
Optional configuration:
Credentials file:
The location of the credentials file where all your configured profiles will be persisted.
Credentials file: /home/test/.aws/credentials
Browse...
Automatically reload accounts when the credentials file is modified in the file system.
Timeouts:
Connection Timeout (ms)
20000
Socket Timeout (ms)
20000
See Network connections for more ways to configure how the AWS Toolkit connects to the Internet.
Get help or provide feedback on the AWS Java Development Forum
Or directly contact us via aws-eclipse-feedback@amazon.com
Restore Defaults
Apply
Cancel
Apply and Close


Package Explorer
LambdaFunctionHandler.java
demo/pom.xml
demo
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Shift+Alt+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete
Remove from Context
Build Path
Source Shift+Alt+S
Refactor Shift+Alt+T
Import...
Export...
Refresh F5
Close Project
Assign Working Sets...
Coverage As
Run As
Debug As
Validate
Restore from Local History...
Maven
Amazon Web Services
Team
Compare With
Configure
Properties Alt+Enter
Deploy Serverless Project...
Run function on AWS Lambda...
Upload function to AWS Lambda...
Debug on AWS SAM Local...
Run on AWS SAM Local...
Console
Progress
[INFO] BUILD SUCCESS
[INFO] Total time: 02:48 min
[INFO] Finished at: 2018-06-18T02:00:18+05:30


Upload Function to AWS Lambda
Select Target Lambda Function
Choose the region and the target AWS Lambda function you want to create or update for your local lambda handler.
aws
Select Lambda Handler and Target Region
Select the Handler:
com.amazonaws.lambda.demo.LambdaFunctionHandler
Select the AWS Region:
EU (Frankfurt)
Select or Create a Lambda Function:
Choose an existing Lambda function:
Not Found
Create a new Lambda function:
MyJavaFunction


Upload Function to AWS Lambda
Function Configuration
Configure this Lambda function and upload to AWS.
aws
Basic Settings
Name:
MyJavaFunction
Description:
The description for the function (optional)
Function Role
Select the IAM role that AWS Lambda can assume to execute the function on your behalf.
IAM Role:
lambda_basic_execution
Create
Function Versioning and Alias
You can publish a new immutable version and an alias to that version whenever you have a
Publish new version
Provide an alias to this new version
Choose an existing function alias:
Not Found
Create a new function alias:
beta
S3 Bucket for Function Code
S3 Bucket:
lambda-function-bucket-eu-central-1-146179
Create
Upload Lambda zip file with encrytion to protect data at rest by using Amazon S3 master-
None
Amazon S3 master-key
AWS KMS master-key
KMS Key:
aws/lambda
Create
Advanced Settings
Memory (MB):
512
Timeout (s):
15
< Back
Next >
Cancel
Finish



	Function name	Description	Runtime	Code size	Last Modified
○	blog-dev-deleteArticle		Node.js 4.3	6.5 MB	Fri Jun 15 2018 17:54:53 GMT+0530 (IST)
○	blog-dev-updateArticle		Node.js 4.3	6.5 MB	Fri Jun 15 2018 17:54:52 GMT+0530 (IST)
○	greeterHelloWorldSkill	Please use alexa-skills-kit-nodejs-factskill from the Serverless Application Repository	Node.js 6.10	179.5 kB	Thu Jun 07 2018 13:11:57 GMT+0530 (IST)
○	HelloWorld	A starter AWS Lambda function.	Python 2.7	259 bytes	Thu Jun 14 2018 16:10:29 GMT+0530 (IST)
○	MyJavaFunction		Java 8	16.1 MB	Mon Jun 18 2018 02:51:07 GMT+0530 (IST)
○	blog-dev-createArticle		Node.js 4.3	6.5 MB	Fri Jun 15 2018 17:54:52 GMT+0530 (IST)
○	blog-dev-readArticle		Node.js 4.3	6.5 MB	Fri Jun 15 2018 17:54:52 GMT+0530 (IST)
○	handsFreeMesssengerSkill	Please use alexa-skills-kit-nodejs-factskill from the Serverless Application Repository	Node.js 6.10	179.5 kB	Thu Jun 07 2018 13:05:02 GMT+0530 (IST)
○	firstLambda1		Node.js 4.3	216 bytes	Wed Jun 06 2018 16:40:40 GMT+0530 (IST)
○	123		Node.js 4.3	229 bytes	Wed Jun 06 2018 16:09:23 GMT+0530 (IST)

Time (UTC +00:00)	Message
2018-06-17	
	No older events found at the moment. Retry.
21:21:08	START RequestId: 599b9004-7274-11e8-b84f-fd2e27562690 Version: $LATEST
21:21:08	Input: Hello World
21:21:08	END RequestId: 599b9004-7274-11e8-b84f-fd2e27562690
21:21:08	REPORT RequestId: 599b9004-7274-11e8-b84f-fd2e27562690 Duration: 49.79 ms Billed Duration: 100 ms Memory Size: 512 MB Max Memory Used: 41 MB
21:31:08	START RequestId: bf657ee4-7275-11e8-a86c-bfea079c3265 Version: $LATEST
21:31:08	Input: random string
21:31:08	END RequestId: bf657ee4-7275-11e8-a86c-bfea079c3265
21:31:08	REPORT RequestId: bf657ee4-7275-11e8-a86c-bfea079c3265 Duration: 2.29 ms Billed Duration: 100 ms Memory Size: 512 MB Max Memory Used: 41 MB
	No newer events found at the moment. Retry.

LambdaFunctionHandler.java | demo/pom.xml | LambdaFunctionHandlerTest.java

```java
package com.amazonaws.lambda.demo;

import com.amazonaws.services.lambda.runtime.Context;

public class LambdaFunctionHandler implements RequestHandler<String, String> {

    @Override
    public String handleRequest(String input, Context context) {
        context.getLogger().log("Input: " + input);

        // TODO: implement your handler
        return "Remaining time [ms]: " + context.getRemainingTimeInMillis();
    }
}
```

Problems | Javadoc | Declaration | Console | Progress

```
com.amazonaws.lambda.demo.LambdaFunctionHandler Lambda Console
Uploading function code to MyJavaFunction...
Upload success. Function ARN: arn:aws:lambda:eu-central-1:019859648260:function:MyJavaFunction
Invoking function...
=================== FUNCTION OUTPUT ===================
"Remaining time [ms]: 14966"
=================== FUNCTION LOG OUTPUT ===================
START RequestId: 84fab445-7276-11e8-88bc-bdf9fa9c4f31 Version: $LATEST
Input: random stringEND RequestId: 84fab445-7276-11e8-88bc-bdf9fa9c4f31
REPORT RequestId: 84fab445-7276-11e8-88bc-bdf9fa9c4f31  Duration: 37.89 ms      Billed Duration: 100 ms         Memory Size: 512 MB     Max Memory Used: 41 MB
```

Create a new AWS Lambda function

Create a new AWS Lambda function in the workspace

Source folder: demo/src/main/java [Browse...]

Package: com.amazonaws.lambda.demo [Browse...]

Name: S3FunctionHandler

Lambda Function Handler

Each Lambda function must specify a handler class which the service will use as the entry point to begin execution. Learn more about Lambda Java function handler.

Input Type: S3 Event

An Amazon S3 trigger that retrieves metadata for the object that has been updated.

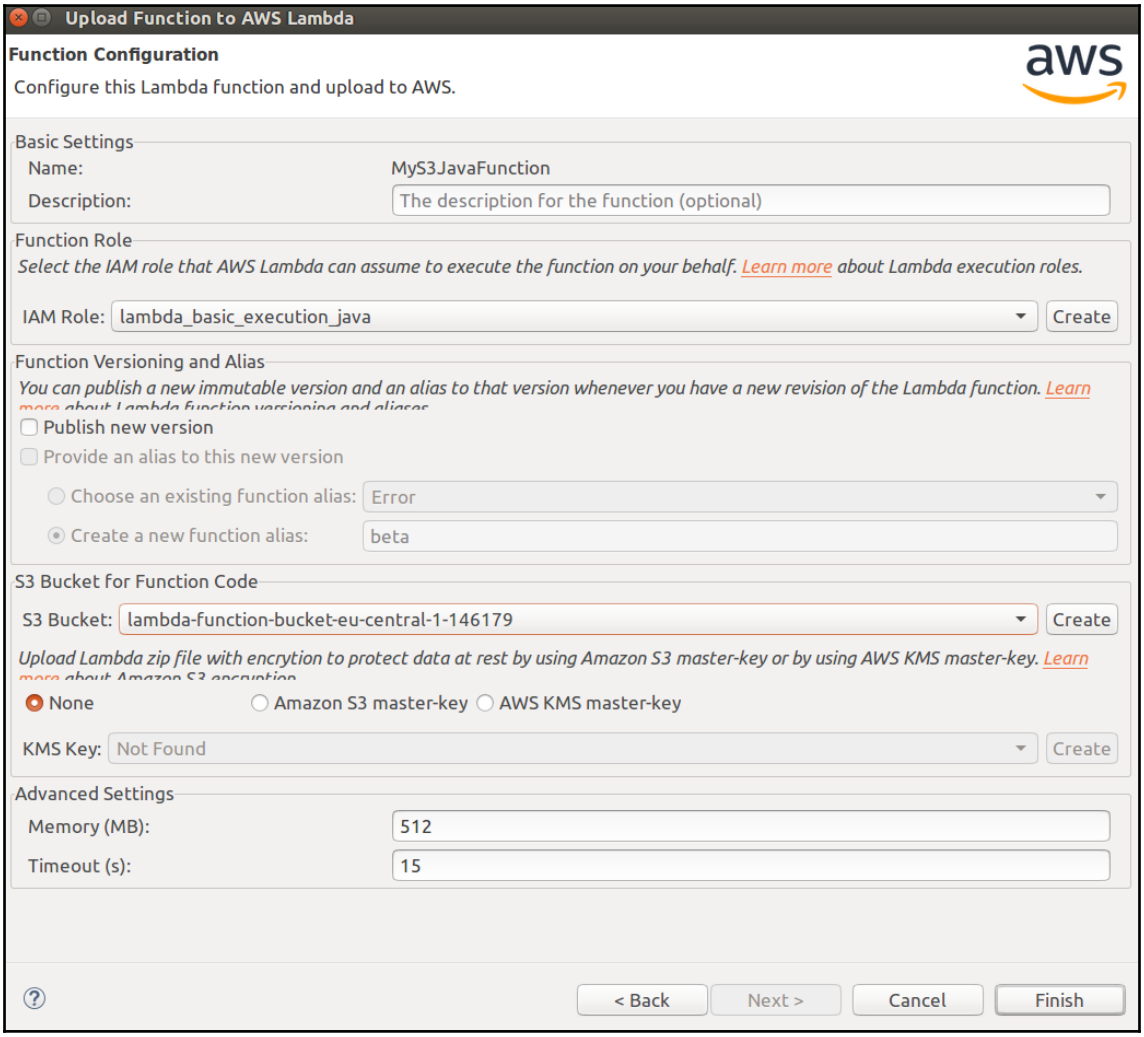
Preview:

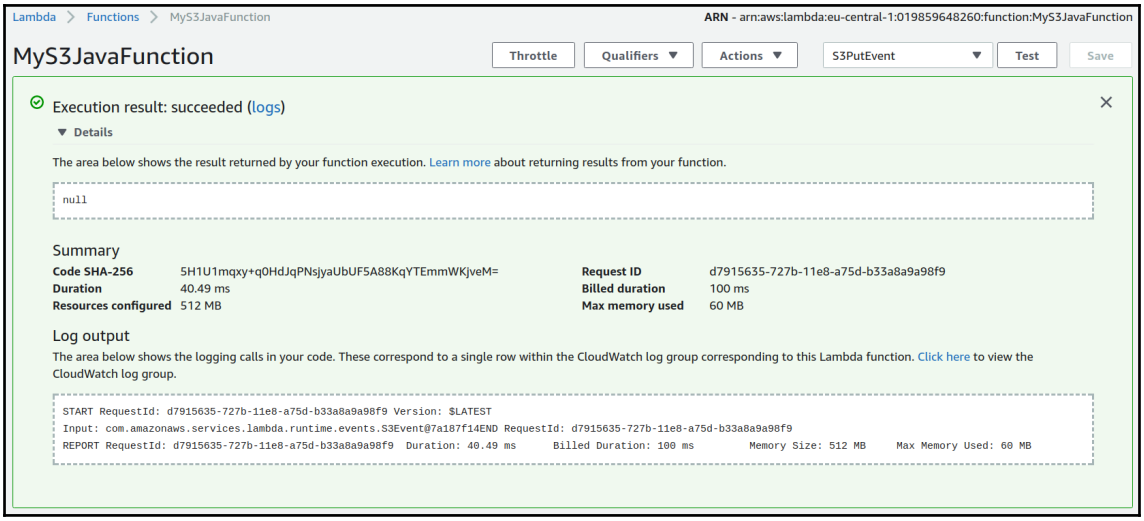
```java
package com.amazonaws.lambda.demo;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.S3Event;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.S3Object;

public class S3FunctionHandler implements RequestHandler<S3Event, String> {

    private AmazonS3 s3 = AmazonS3ClientBuilder.standard().build();

    public S3FunctionHandler() {}

    // Test purpose only.
    S3FunctionHandler(AmazonS3 s3) {
        this.s3 = s3;
    }
```

[Cancel] [Finish]

LambdaFunctionHandler.java | S3FunctionHandler.java

```java
package com.amazonaws.lambda.demo;

import com.amazonaws.services.lambda.runtime.Context;

public class S3FunctionHandler implements RequestHandler<S3Event, Object> {

    @Override
    public Object handleRequest(S3Event input, Context context) {
        context.getLogger().log("Input: " + input);

        // TODO: implement your handler
        return null;
    }
}
```

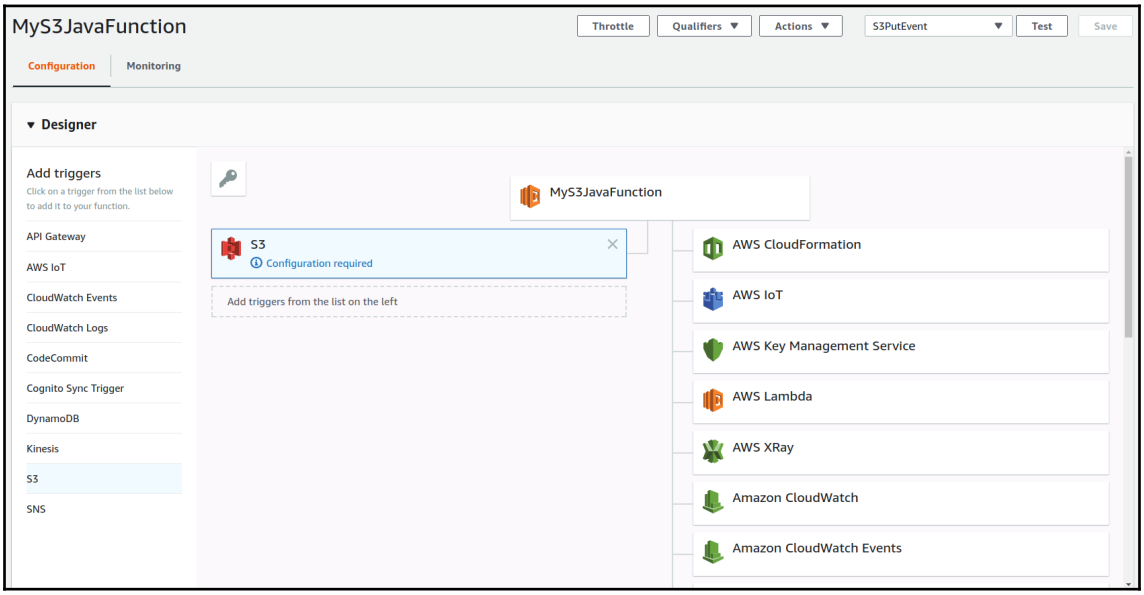


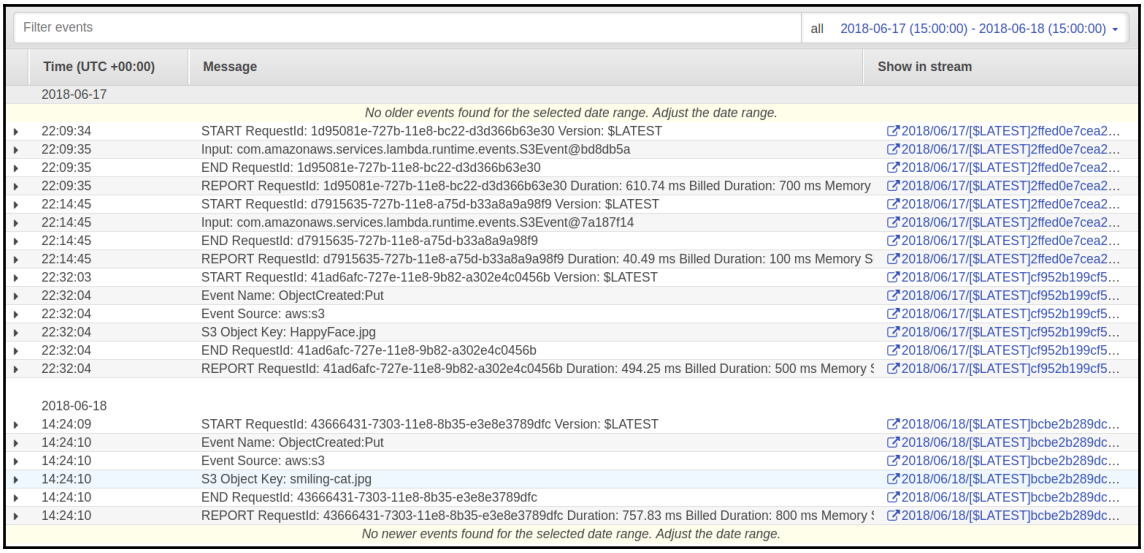


LambdaFunctionHandler.java | S3FunctionHandler.java

```java
package com.amazonaws.lambda.demo;

import com.amazonaws.services.lambda.runtime.Context;

public class S3FunctionHandler implements RequestHandler<S3Event, Object> {

    @Override
    public Object handleRequest(S3Event input, Context context) {
        for(S3EventNotificationRecord rec : input.getRecords()) {
            context.getLogger().log("Event Name: " + rec.getEventName() + "\n");
            context.getLogger().log("Event Source: " + rec.getEventSource() + "\n");
            S3ObjectEntity s3object = rec.getS3().getObject();
            context.getLogger().log("S3 Object Key: " + s3object.getKey() + "\n");
        }

        // TODO: implement your handler
        return null;
    }
}
```


MyS3JavaFunction
Throttle
Qualifiers
Actions
S3PutEvent
Test
Save
Log output
The area 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: 41ad6afc-727e-11e8-9b82-a302e4c0456b Version: $LATEST
Event Name: ObjectCreated:Put
Event Source: aws:s3
S3 Object Key: HappyFace.jpg
END RequestId: 41ad6afc-727e-11e8-9b82-a302e4c0456b
REPORT RequestId: 41ad6afc-727e-11e8-9b82-a302e4c0456b Duration: 494.25 ms Billed Duration: 500 ms Memory Size: 512 MB
Max Memory Used: 58 MB

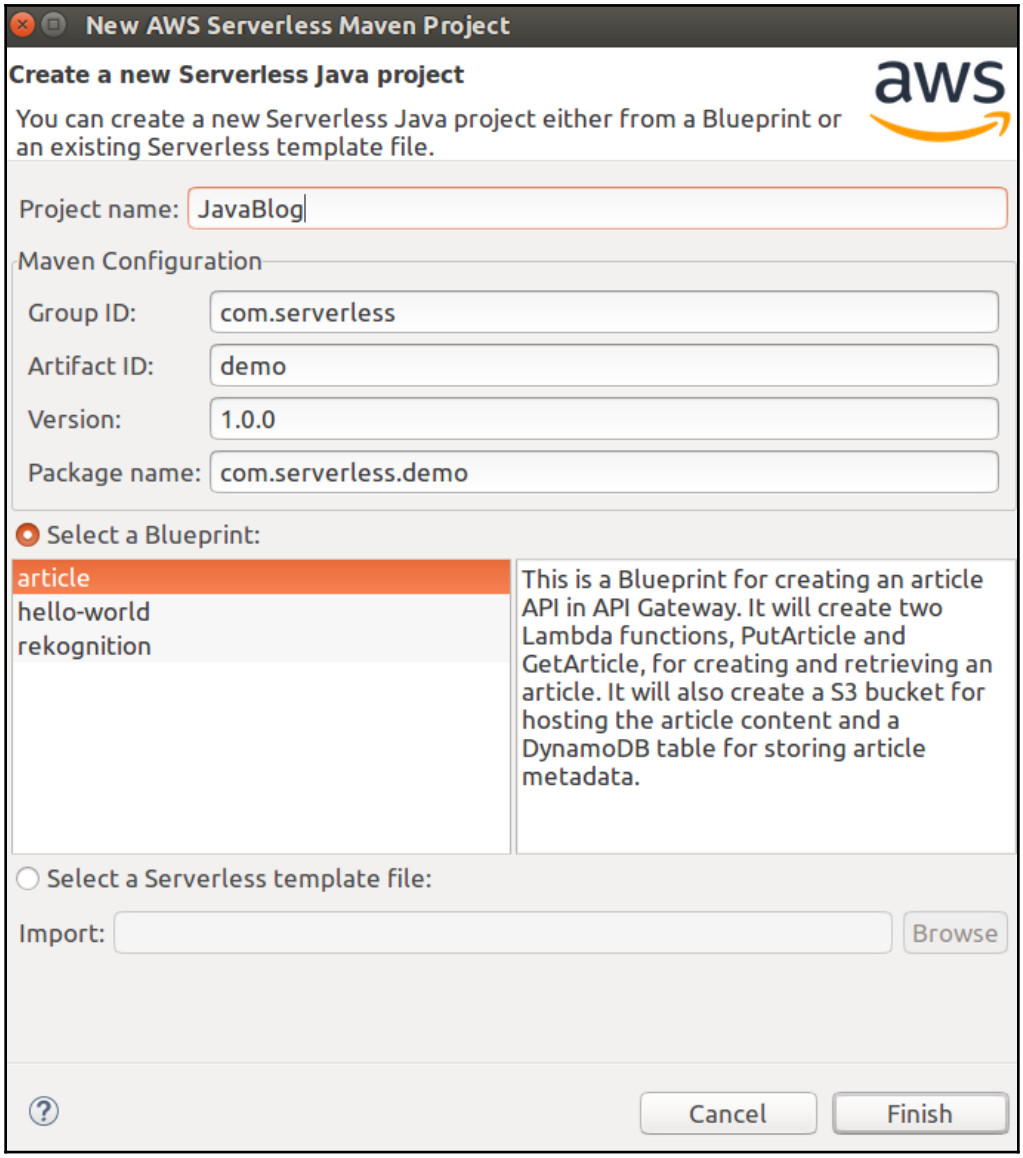

MyS3JavaFunction
Throttle
Qualifiers
Actions
S3PutEvent
Test
Save
Configuration
Monitoring
Designer
Add triggers
Click on a trigger from the list below to add it to your function.
API Gateway
AWS IoT
CloudWatch Events
CloudWatch Logs
CodeCommit
Cognito Sync Trigger
DynamoDB
Kinesis
S3
SNS
MyS3JavaFunction
S3
Configuration required
Add triggers from the list on the left
AWS CloudFormation
AWS IoT
AWS Key Management Service
AWS Lambda
AWS XRay
Amazon CloudWatch
Amazon CloudWatch Events



Filter events

all 2018-06-17 (15:00:00) - 2018-06-18 (15:00:00)

Time (UTC +00:00)	Message	Show in stream
2018-06-17		
No older events found for the selected date range. Adjust the date range.		
22:09:34	START RequestId: 1d95081e-727b-11e8-bc22-d3d366b63e30 Version: $LATEST	2018/06/17/[$LATEST]2ffed0e7cea2…
22:09:35	Input: com.amazonaws.services.lambda.runtime.events.S3Event@bd8db5a	2018/06/17/[$LATEST]2ffed0e7cea2…
22:09:35	END RequestId: 1d95081e-727b-11e8-bc22-d3d366b63e30	2018/06/17/[$LATEST]2ffed0e7cea2…
22:09:35	REPORT RequestId: 1d95081e-727b-11e8-bc22-d3d366b63e30 Duration: 610.74 ms Billed Duration: 700 ms Memory	2018/06/17/[$LATEST]2ffed0e7cea2…
22:14:45	START RequestId: d7915635-727b-11e8-a75d-b33a8a9a98f9 Version: $LATEST	2018/06/17/[$LATEST]2ffed0e7cea2…
22:14:45	Input: com.amazonaws.services.lambda.runtime.events.S3Event@7a187f14	2018/06/17/[$LATEST]2ffed0e7cea2…
22:14:45	END RequestId: d7915635-727b-11e8-a75d-b33a8a9a98f9	2018/06/17/[$LATEST]2ffed0e7cea2…
22:14:45	REPORT RequestId: d7915635-727b-11e8-a75d-b33a8a9a98f9 Duration: 40.49 ms Billed Duration: 100 ms Memory S	2018/06/17/[$LATEST]2ffed0e7cea2…
22:32:03	START RequestId: 41ad6afc-727e-11e8-9b82-a302e4c0456b Version: $LATEST	2018/06/17/[$LATEST]cf952b199cf5…
22:32:04	Event Name: ObjectCreated:Put	2018/06/17/[$LATEST]cf952b199cf5…
22:32:04	Event Source: aws:s3	2018/06/17/[$LATEST]cf952b199cf5…
22:32:04	S3 Object Key: HappyFace.jpg	2018/06/17/[$LATEST]cf952b199cf5…
22:32:04	END RequestId: 41ad6afc-727e-11e8-9b82-a302e4c0456b	2018/06/17/[$LATEST]cf952b199cf5…
22:32:04	REPORT RequestId: 41ad6afc-727e-11e8-9b82-a302e4c0456b Duration: 494.25 ms Billed Duration: 500 ms Memory S	2018/06/17/[$LATEST]cf952b199cf5…
2018-06-18		
14:24:09	START RequestId: 43666431-7303-11e8-8b35-e3e8e3789dfc Version: $LATEST	2018/06/18/[$LATEST]bcbe2b289dc…
14:24:10	Event Name: ObjectCreated:Put	2018/06/18/[$LATEST]bcbe2b289dc…
14:24:10	Event Source: aws:s3	2018/06/18/[$LATEST]bcbe2b289dc…
14:24:10	S3 Object Key: smiling-cat.jpg	2018/06/18/[$LATEST]bcbe2b289dc…
14:24:10	END RequestId: 43666431-7303-11e8-8b35-e3e8e3789dfc	2018/06/18/[$LATEST]bcbe2b289dc…
14:24:10	REPORT RequestId: 43666431-7303-11e8-8b35-e3e8e3789dfc Duration: 757.83 ms Billed Duration: 800 ms Memory S	2018/06/18/[$LATEST]bcbe2b289dc…
No newer events found for the selected date range. Adjust the date range.		

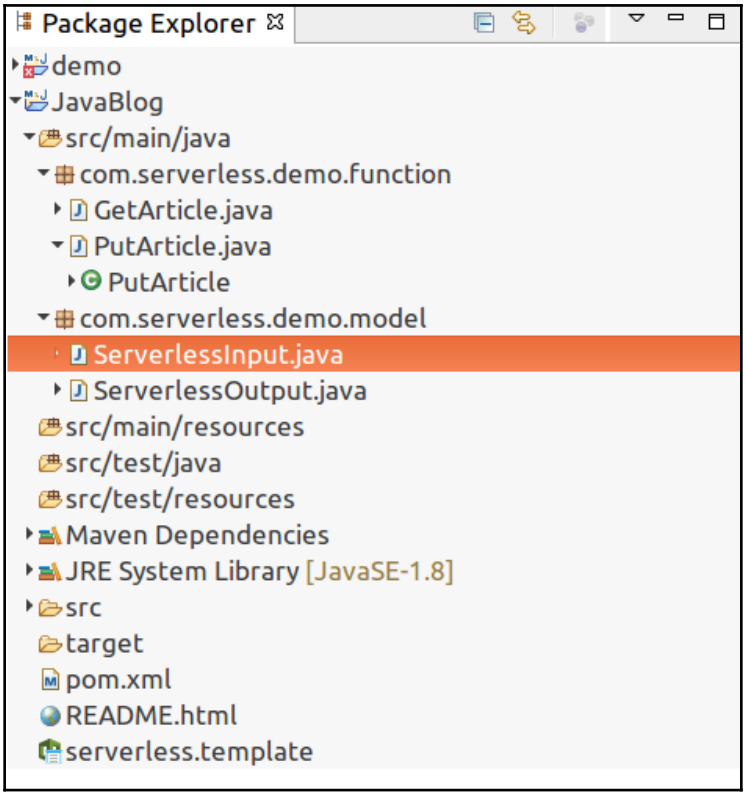

New AWS Serverless Maven Project
Create a new Serverless Java project
You can create a new Serverless Java project either from a Blueprint or an existing Serverless template file.
aws
Project name: JavaBlog
Maven Configuration
Group ID: com.serverless
Artifact ID: demo
Version: 1.0.0
Package name: com.serverless.demo
Select a Blueprint:
article
hello-world
rekognition
This is a Blueprint for creating an article API in API Gateway. It will create two Lambda functions, PutArticle and GetArticle, for creating and retrieving an article. It will also create a S3 bucket for hosting the article content and a DynamoDB table for storing article metadata.
Select a Serverless template file:
Import:
Browse
Cancel
Finish

PutArticle.java

```java
package com.serverless.demo.function;

import java.io.ByteArrayInputStream;

/**
 * Lambda function that triggered by the API Gateway event "POST /". It reads all the query parameters as the metadata for this
 * article and stores them to a DynamoDB table. It reads the payload as the content of the article and stores it to a S3 bucket.
 */
public class PutArticle implements RequestHandler<ServerlessInput, ServerlessOutput> {
    // DynamoDB table name for storing article metadata.
    private static final String ARTICLE_TABLE_NAME = System.getenv("ARTICLE_TABLE_NAME");
    // DynamoDB table attribute name for storing article id.
    private static final String ARTICLE_TABLE_ID_NAME = "id";
    // DynamoDB table attribute name for storing the bucket name where holds the article's content.
    private static final String ARTICLE_TABLE_BUCKET_NAME = "bucket";
    // DynamoDB table attribute name for storing the bucket object key name that contains the article's content.
    private static final String ARTICLE_TABLE_KEY_NAME = "key";
    // S3 bucket name for storing article content.
    private static final String ARTICLE_BUCKET_NAME = System.getenv("ARTICLE_BUCKET_NAME");
    @Override
    public ServerlessOutput handleRequest(ServerlessInput serverlessInput, Context context) {
            // Using builder to create the clients could allow us to dynamically load the region from the AWS_REGION environment
            // variable. Therefore we can deploy the Lambda functions to different regions without code change.
            AmazonDynamoDB dynamoDb = AmazonDynamoDBClientBuilder.standard().build();
            AmazonS3 s3 = AmazonS3ClientBuilder.standard().build();
            ServerlessOutput output = new ServerlessOutput();

            try {
                String keyName = UUID.randomUUID().toString();
                String content = serverlessInput.getBody();
                s3.putObject(new PutObjectRequest(
                        ARTICLE_BUCKET_NAME,
                        keyName,
                        new ByteArrayInputStream(content.getBytes(StandardCharsets.UTF_8)),
                        new ObjectMetadata())
                );

                Map<String, AttributeValue> attributes = convert(serverlessInput.getQueryStringParameters());
                attributes.putIfAbsent(ARTICLE_TABLE_ID_NAME, new AttributeValue().withS(UUID.randomUUID().toString()));
                attributes.put(ARTICLE_TABLE_BUCKET_NAME, new AttributeValue().withS(ARTICLE_BUCKET_NAME));
                attributes.put(ARTICLE_TABLE_KEY_NAME, new AttributeValue().withS(keyName));
                dynamoDb.putItem(new PutItemRequest()
                        .withTableName(ARTICLE_TABLE_NAME)
                        .withItem(attributes));
                output.setStatusCode(200);
```

PutArticle.java

```java
    // DynamoDB table name for storing article metadata.
    private static final String ARTICLE_TABLE_NAME = System.getenv("ARTICLE_TABLE_NAME");
    // DynamoDB table attribute name for storing article id.
    private static final String ARTICLE_TABLE_ID_NAME = "id";
    // DynamoDB table attribute name for storing the bucket name where holds the article's content.
    private static final String ARTICLE_TABLE_BUCKET_NAME = "bucket";
    // DynamoDB table attribute name for storing the bucket object key name that contains the article's content.
    private static final String ARTICLE_TABLE_KEY_NAME = "key";
    // S3 bucket name for storing article content.
    private static final String ARTICLE_BUCKET_NAME = System.getenv("ARTICLE_BUCKET_NAME");
    @Override
    public ServerlessOutput handleRequest(ServerlessInput serverlessInput, Context context) {
            // Using builder to create the clients could allow us to dynamically load the region from the AWS_REGION envir
            // variable. Therefore we can deploy the Lambda functions to different regions without code change.
            AmazonDynamoDB dynamoDb = AmazonDynamoDBClientBuilder.standard().build();
            AmazonS3 s3 = AmazonS3ClientBuilder.standard().build();
            ServerlessOutput output = new ServerlessOutput();

            try {
                String keyName = UUID.randomUUID().toString();
                String content = serverlessInput.getBody();
                s3.putObject(new PutObjectRequest(
                        ARTICLE_BUCKET_NAME,
                        keyName,
                        new ByteArrayInputStream(content.getBytes(StandardCharsets.UTF_8)),
                        new ObjectMetadata())
                );

                Map<String, AttributeValue> attributes = convert(serverlessInput.getQueryStringParameters());
                attributes.putIfAbsent(ARTICLE_TABLE_ID_NAME, new AttributeValue().withS(UUID.randomUUID().toString()));
                attributes.put(ARTICLE_TABLE_BUCKET_NAME, new AttributeValue().withS(ARTICLE_BUCKET_NAME));
                attributes.put(ARTICLE_TABLE_KEY_NAME, new AttributeValue().withS(keyName));
                dynamoDb.putItem(new PutItemRequest()
```


Package Explorer
demo
JavaBlog
src/main/java
com.serverless.demo.function
GetArticle.java
PutArticle.java
PutArticle
com.serverless.demo.model
ServerlessInput.java
ServerlessOutput.java
src/main/resources
src/test/java
src/test/resources
Maven Dependencies
JRE System Library [JavaSE-1.8]
src
target
pom.xml
README.html
serverless.template

PutArticle.java | ServerlessInput.java

```java
package com.serverless.demo.model;

import java.util.Map;

public class ServerlessInput {

    private String resource;
    private String path;
    private String httpMethod;
    private Map<String, String> headers;
    private Map<String, String> queryStringParameters;
    private Map<String, String> pathParameters;
    private Map<String, String> stageVariables;
    private String body;
    private RequestContext requestContext;
    private Boolean isBase64Encoded;

    public String getResource() {
        return resource;
    }
    public void setResource(String resource) {
        this.resource = resource;
    }
    public ServerlessInput withResource(String resource) {
        setResource(resource);
        return this;
    }
    public String getPath() {
        return path;
    }
    public void setPath(String path) {
        this.path = path;
    }
    public ServerlessInput withPath(String path) {
        setPath(path);
        return this;
    }
    public String getHttpMethod() {
```

```
42      public ServerlessOutput handleRequest(ServerlessInput serverlessInput, Context context) {
43              // Using builder to create the clients could allow us to dynamically load the region from the AWS_REGION envi
44              // variable. Therefore we can deploy the Lambda functions to different regions without code change.
45              AmazonDynamoDB dynamoDb = AmazonDynamoDBClientBuilder.standard().build();
46              AmazonS3 s3 = AmazonS3ClientBuilder.standard().build();
47              ServerlessOutput output = new ServerlessOutput();
48
49              try {
50                  String keyName = UUID.randomUUID().toString();
51                  String content = serverlessInput.getBody();
52                  s3.putObject(new PutObjectRequest(
53                          ARTICLE_BUCKET_NAME,
54                          keyName,
55                          new ByteArrayInputStream(content.getBytes(StandardCharsets.UTF_8)),
56                          new ObjectMetadata())
57                  );
58
59                  Map<String, AttributeValue> attributes = convert(serverlessInput.getQueryStringParameters());
60                  attributes.putIfAbsent(ARTICLE_TABLE_ID_NAME, new AttributeValue().withS(UUID.randomUUID().toString()));
61                  attributes.put(ARTICLE_TABLE_BUCKET_NAME, new AttributeValue().withS(ARTICLE_BUCKET_NAME));
62                  attributes.put(ARTICLE_TABLE_KEY_NAME, new AttributeValue().withS(keyName));
63                  dynamoDb.putItem(new PutItemRequest()
64                          .withTableName(ARTICLE_TABLE_NAME)
```

```
AmazonDynamoDB dynamoDb = AmazonDynamoDBClientBuilder.standard().build();
AmazonS3 s3 = AmazonS3ClientBuilder.standard().build();
```

```
String keyName = UUID.randomUUID().toString();
String content = serverlessInput.getBody();
s3.putObject(new PutObjectRequest(
        ARTICLE_BUCKET_NAME,
        keyName,
        new ByteArrayInputStream(content.getBytes(StandardCharsets.UTF_8)),
        new ObjectMetadata())
);
```

```
Map<String, AttributeValue> attributes = convert(serverlessInput.getQueryStringParameters());
attributes.putIfAbsent(ARTICLE_TABLE_ID_NAME, new AttributeValue().withS(UUID.randomUUID().toString()));
attributes.put(ARTICLE_TABLE_BUCKET_NAME, new AttributeValue().withS(ARTICLE_BUCKET_NAME));
attributes.put(ARTICLE_TABLE_KEY_NAME, new AttributeValue().withS(keyName));
dynamoDb.putItem(new PutItemRequest()
        .withTableName(ARTICLE_TABLE_NAME)
        .withItem(attributes));
```

```
output.setStatusCode(200);
output.setBody("Successfully inserted article " + attributes.get(ARTICLE_TABLE_ID_NAME).getS());
```

```
if (serverlessInput.getQueryStringParameters() == null || serverlessInput.getQueryStringParameters().get(ARTICLE_TABLE_ID_NAME) == null) {
        throw new Exception("Parameter " + ARTICLE_TABLE_ID_NAME + " in query must be provided!");
}
```

```
Map<String, AttributeValue> item = dynamoDb.getItem(new GetItemRequest()
        .withTableName(ARTICLE_TABLE_NAME)
        .withKey(key))
        .getItem();
```

```
String s3ObjectKey = item.get(ARTICLE_TABLE_KEY_NAME).getS();
```

```
"Description": "Simple article service.",
"Parameters" : {
  "ArticleBucketName" : {
      "Type" : "String",
      "Default": "serverless-blueprint-article-bucket",
      "Description" : "Name of S3 bucket used to store the article content. If left blank, AWS CloudFormation would manage this resource.",
      "MinLength" : "0"
  },
  "ArticleTableName" : {
      "Type" : "String",
      "Default": "serverless-blueprint-article-table",
      "Description" : "Name of DynamoDB table used to store the article metadata. If left blank, AWS CloudFormation would manage this resource.",
      "MinLength" : "0"
    },
  "ReadCapacity" : {
      "Type" : "Number",
      "Description" : "Read capacity for the DynamoDB blog table.",
      "Default" : "3",
      "MinValue" : 1
  },
  "WriteCapacity" : {
      "Type" : "Number",
      "Description" : "Write capacity for the DynamoDB blog table.",
      "Default" : "3",
      "MinValue" : 1
  }
},
```

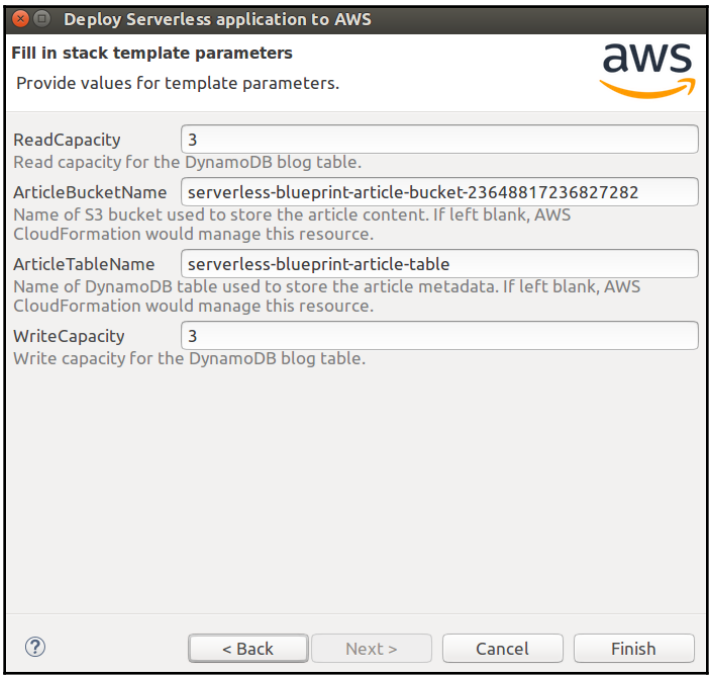
```
"GetArticle": {
  "Type": "AWS::Serverless::Function",
  "Properties": {
    "Handler": "com.serverless.demo.function.GetArticle",
    "Runtime" : "java8",
    "CodeUri" : "./target/demo-1.0.0.jar",
    "Policies": [
      "AmazonDynamoDBReadOnlyAccess",
      "AmazonS3ReadOnlyAccess"
    ],
    "Environment" : {
      "Variables" : {
        "ARTICLE_TABLE_NAME" : { "Ref" : "ArticleTableName" },
        "ARTICLE_BUCKET_NAME" : { "Ref" : "ArticleBucketName" }
      }
    },
    "Events": {
      "GetResource": {
        "Type": "Api",
        "Properties": {
          "Path": "/",
          "Method": "get"
        }
      }
    }
  }
```

```
      "Policies": [
        "AmazonDynamoDBReadOnlyAccess",
        "AmazonS3ReadOnlyAccess"
      ],
      "Environment" : {
        "Variables" : {
          "ARTICLE_TABLE_NAME" : { "Ref" : "ArticleTableName" },
          "ARTICLE_BUCKET_NAME" : { "Ref" : "ArticleBucketName" }
        }
      },
```

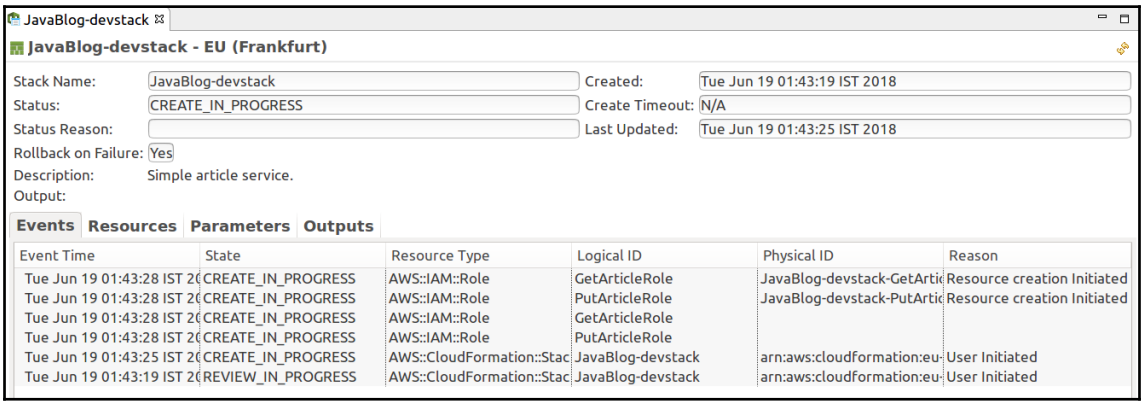
```
"PutArticle": {
  "Type": "AWS::Serverless::Function",
  "Properties": {
    "Handler": "com.serverless.demo.function.PutArticle",
    "Runtime" : "java8",
    "CodeUri" : "./target/demo-1.0.0.jar",
    "Policies": [
      "AmazonDynamoDBFullAccess",
      "AmazonS3FullAccess"
    ],
```

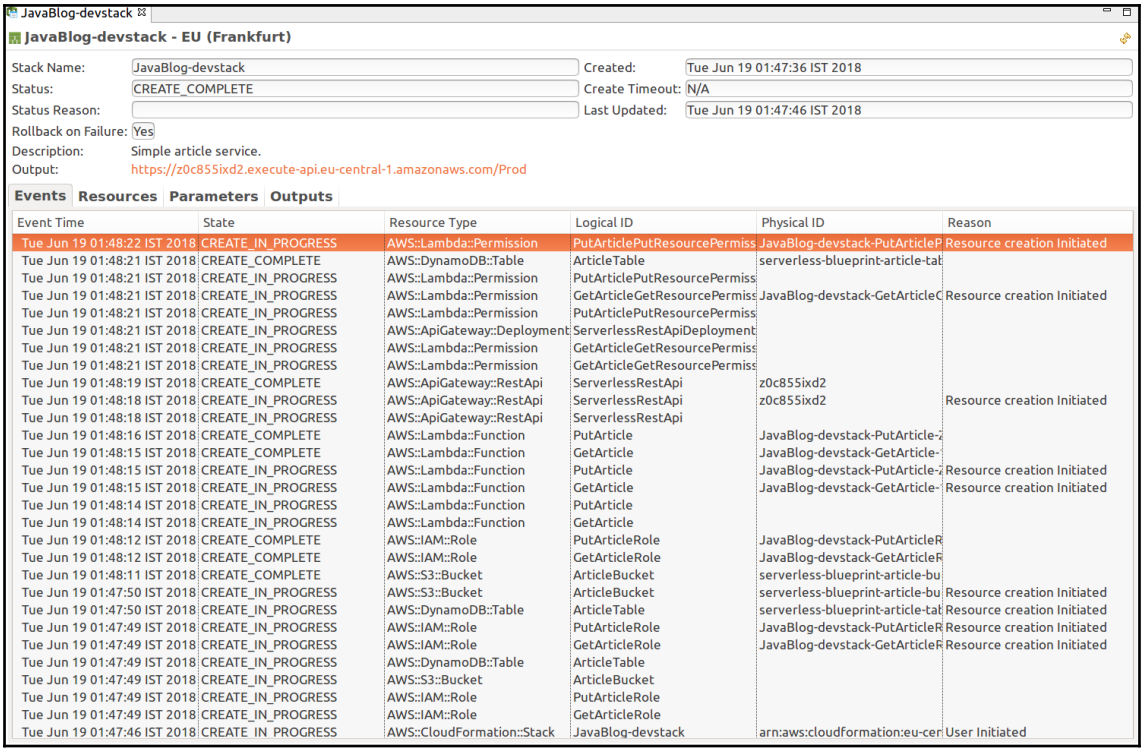
```
  "ArticleTable": {
    "Type": "AWS::DynamoDB::Table",
    "Properties": {
      "AttributeDefinitions": [
        {
          "AttributeName": "id",
          "AttributeType": "S"
        }
      ],
      "KeySchema": [
        {
          "AttributeName": "id",
          "KeyType": "HASH"
        }
      ],
      "ProvisionedThroughput": {
        "ReadCapacityUnits": {"Ref" : "ReadCapacity"},
        "WriteCapacityUnits": {"Ref" : "WriteCapacity"}
      },
      "TableName": {"Ref" : "ArticleTableName"}
    }
  },
```

```
  "ArticleBucket": {
    "Type": "AWS::S3::Bucket",
    "Properties": {
      "BucketName": {"Ref" : "ArticleBucketName"}
    }
  }
```




```
"Default": "serverless-blueprint-article-bucket-23648817236827282",
```





JavaBlog-devstack

JavaBlog-devstack - EU (Frankfurt)

Stack Name:	JavaBlog-devstack	Created:	Tue Jun 19 01:47:36 IST 2018
Status:	CREATE_COMPLETE	Create Timeout:	N/A
Status Reason:		Last Updated:	Tue Jun 19 01:47:46 IST 2018

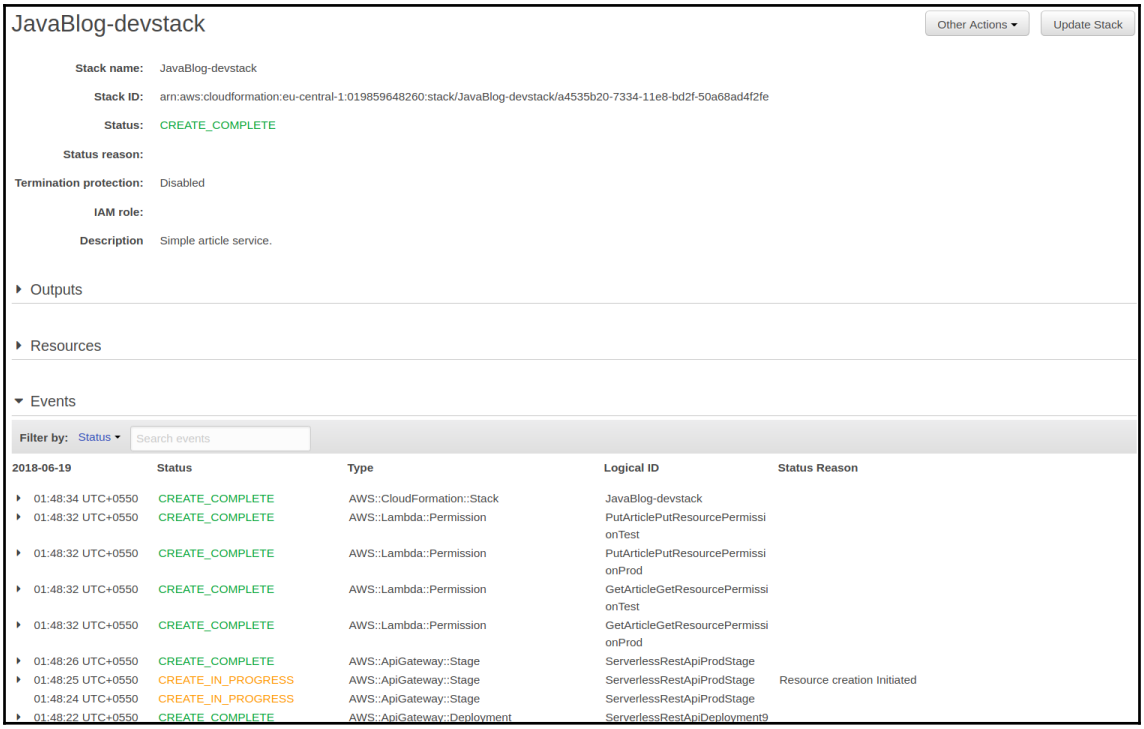
Rollback on Failure: Yes

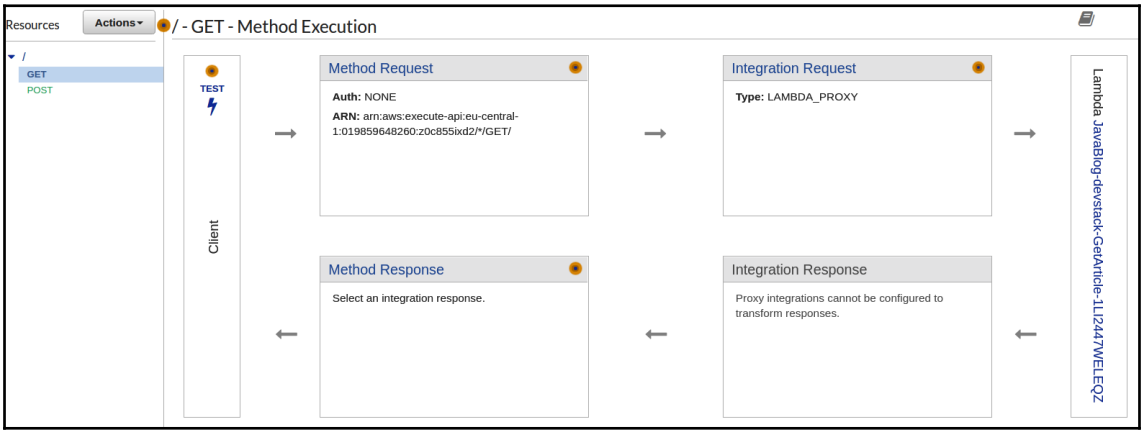
Description: Simple article service.

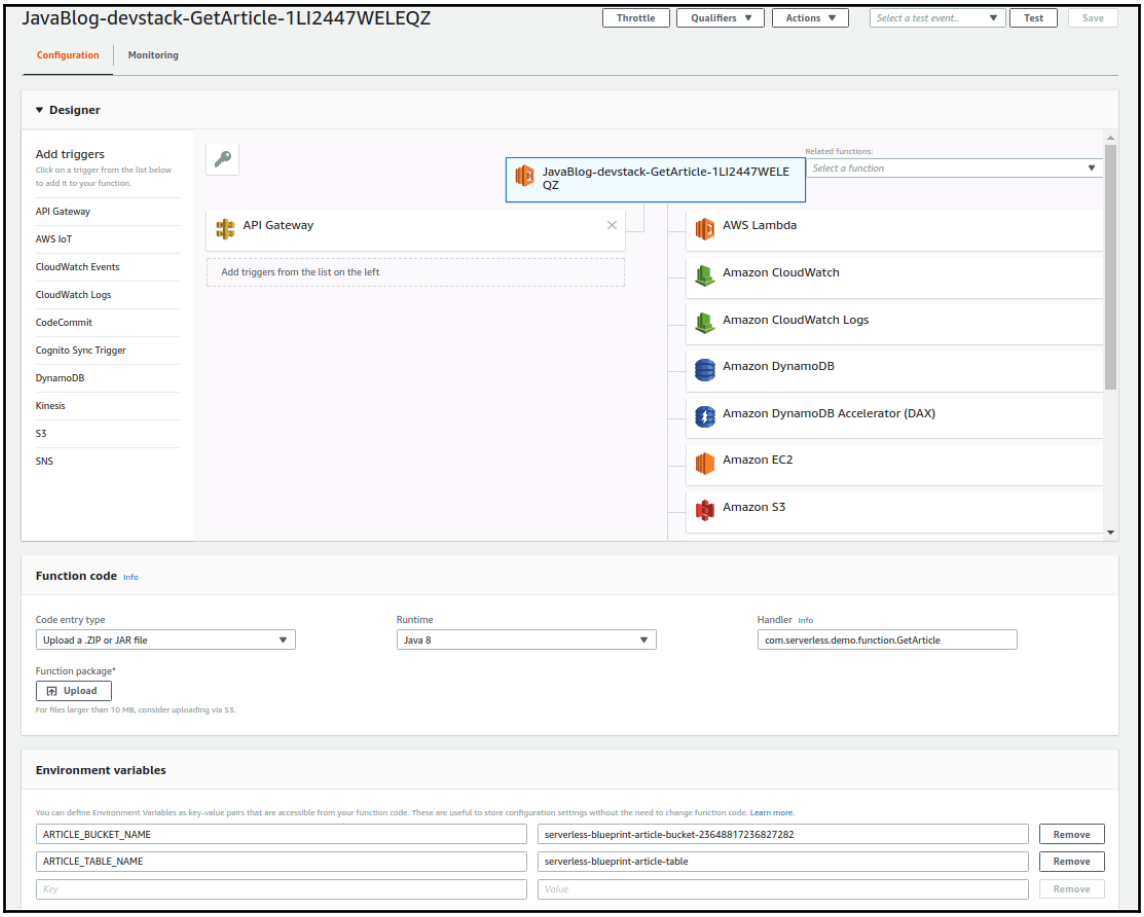
Output: https://z0c855ixd2.execute-api.eu-central-1.amazonaws.com/Prod

Events | **Resources** | **Parameters** | **Outputs**

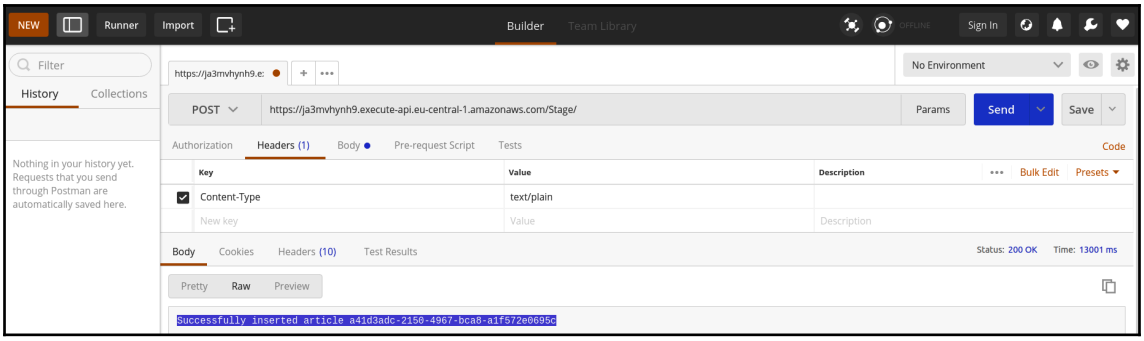
Event Time	State	Resource Type	Logical ID	Physical ID	Reason
Tue Jun 19 01:48:22 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Permission	PutArticlePutResourcePermiss	JavaBlog-devstack-PutArticleP	Resource creation Initiated
Tue Jun 19 01:48:21 IST 2018	CREATE_COMPLETE	AWS::DynamoDB::Table	ArticleTable	serverless-blueprint-article-tat	
Tue Jun 19 01:48:21 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Permission	PutArticlePutResourcePermiss		
Tue Jun 19 01:48:21 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Permission	GetArticleGetResourcePermiss	JavaBlog-devstack-GetArticleC	Resource creation Initiated
Tue Jun 19 01:48:21 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Permission	PutArticlePutResourcePermiss		
Tue Jun 19 01:48:21 IST 2018	CREATE_IN_PROGRESS	AWS::ApiGateway::Deployment	ServerlessRestApiDeployment		
Tue Jun 19 01:48:21 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Permission	GetArticleGetResourcePermiss		
Tue Jun 19 01:48:21 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Permission	GetArticleGetResourcePermiss		
Tue Jun 19 01:48:19 IST 2018	CREATE_COMPLETE	AWS::ApiGateway::RestApi	ServerlessRestApi	z0c855ixd2	
Tue Jun 19 01:48:18 IST 2018	CREATE_IN_PROGRESS	AWS::ApiGateway::RestApi	ServerlessRestApi	z0c855ixd2	Resource creation Initiated
Tue Jun 19 01:48:18 IST 2018	CREATE_IN_PROGRESS	AWS::ApiGateway::RestApi	ServerlessRestApi		
Tue Jun 19 01:48:16 IST 2018	CREATE_COMPLETE	AWS::Lambda::Function	PutArticle	JavaBlog-devstack-PutArticle-A	
Tue Jun 19 01:48:15 IST 2018	CREATE_COMPLETE	AWS::Lambda::Function	GetArticle	JavaBlog-devstack-GetArticle-	
Tue Jun 19 01:48:15 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Function	PutArticle	JavaBlog-devstack-PutArticle-A	Resource creation Initiated
Tue Jun 19 01:48:15 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Function	GetArticle	JavaBlog-devstack-GetArticle-	Resource creation Initiated
Tue Jun 19 01:48:14 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Function	PutArticle		
Tue Jun 19 01:48:14 IST 2018	CREATE_IN_PROGRESS	AWS::Lambda::Function	GetArticle		
Tue Jun 19 01:48:12 IST 2018	CREATE_COMPLETE	AWS::IAM::Role	PutArticleRole	JavaBlog-devstack-PutArticleR	
Tue Jun 19 01:48:12 IST 2018	CREATE_COMPLETE	AWS::IAM::Role	GetArticleRole	JavaBlog-devstack-GetArticleR	
Tue Jun 19 01:48:11 IST 2018	CREATE_COMPLETE	AWS::S3::Bucket	ArticleBucket	serverless-blueprint-article-bu	
Tue Jun 19 01:47:50 IST 2018	CREATE_IN_PROGRESS	AWS::S3::Bucket	ArticleBucket	serverless-blueprint-article-bu	Resource creation Initiated
Tue Jun 19 01:47:50 IST 2018	CREATE_IN_PROGRESS	AWS::DynamoDB::Table	ArticleTable	serverless-blueprint-article-tat	Resource creation Initiated
Tue Jun 19 01:47:49 IST 2018	CREATE_IN_PROGRESS	AWS::IAM::Role	PutArticleRole	JavaBlog-devstack-PutArticleR	Resource creation Initiated
Tue Jun 19 01:47:49 IST 2018	CREATE_IN_PROGRESS	AWS::IAM::Role	GetArticleRole	JavaBlog-devstack-GetArticleR	Resource creation Initiated
Tue Jun 19 01:47:49 IST 2018	CREATE_IN_PROGRESS	AWS::DynamoDB::Table	ArticleTable		
Tue Jun 19 01:47:49 IST 2018	CREATE_IN_PROGRESS	AWS::S3::Bucket	ArticleBucket		
Tue Jun 19 01:47:49 IST 2018	CREATE_IN_PROGRESS	AWS::IAM::Role	PutArticleRole		
Tue Jun 19 01:47:49 IST 2018	CREATE_IN_PROGRESS	AWS::IAM::Role	GetArticleRole		
Tue Jun 19 01:47:46 IST 2018	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	JavaBlog-devstack	arn:aws:cloudformation:eu-cer	User Initiated

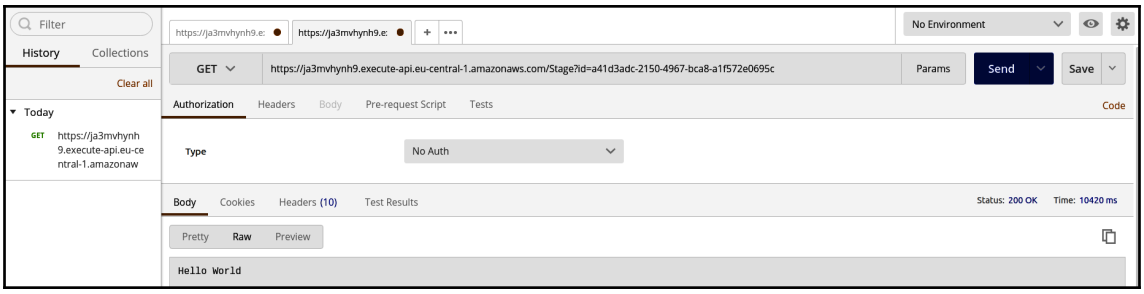

JavaBlog-devstack
Other Actions
Update Stack
Stack name: JavaBlog-devstack
Stack ID: arn:aws:cloudformation:eu-central-1:019859648260:stack/JavaBlog-devstack/a4535b20-7334-11e8-bd2f-50a68ad4f2fe
Status: CREATE_COMPLETE
Status reason:
Termination protection: Disabled
IAM role:
Description Simple article service.
Outputs
Resources
Events
Filter by: Status
Search events
2018-06-19 Status Type Logical ID Status Reason
01:48:34 UTC+0550 CREATE_COMPLETE AWS::CloudFormation::Stack JavaBlog-devstack
01:48:32 UTC+0550 CREATE_COMPLETE AWS::Lambda::Permission PutArticlePutResourcePermissionTest
01:48:32 UTC+0550 CREATE_COMPLETE AWS::Lambda::Permission PutArticlePutResourcePermissionProd
01:48:32 UTC+0550 CREATE_COMPLETE AWS::Lambda::Permission GetArticleGetResourcePermissionTest
01:48:32 UTC+0550 CREATE_COMPLETE AWS::Lambda::Permission GetArticleGetResourcePermissionProd
01:48:26 UTC+0550 CREATE_COMPLETE AWS::ApiGateway::Stage ServerlessRestApiProdStage
01:48:25 UTC+0550 CREATE_IN_PROGRESS AWS::ApiGateway::Stage ServerlessRestApiProdStage Resource creation Initiated
01:48:24 UTC+0550 CREATE_IN_PROGRESS AWS::ApiGateway::Stage ServerlessRestApiProdStage
01:48:22 UTC+0550 CREATE_COMPLETE AWS::ApiGateway::Deployment ServerlessRestApiDeployment9

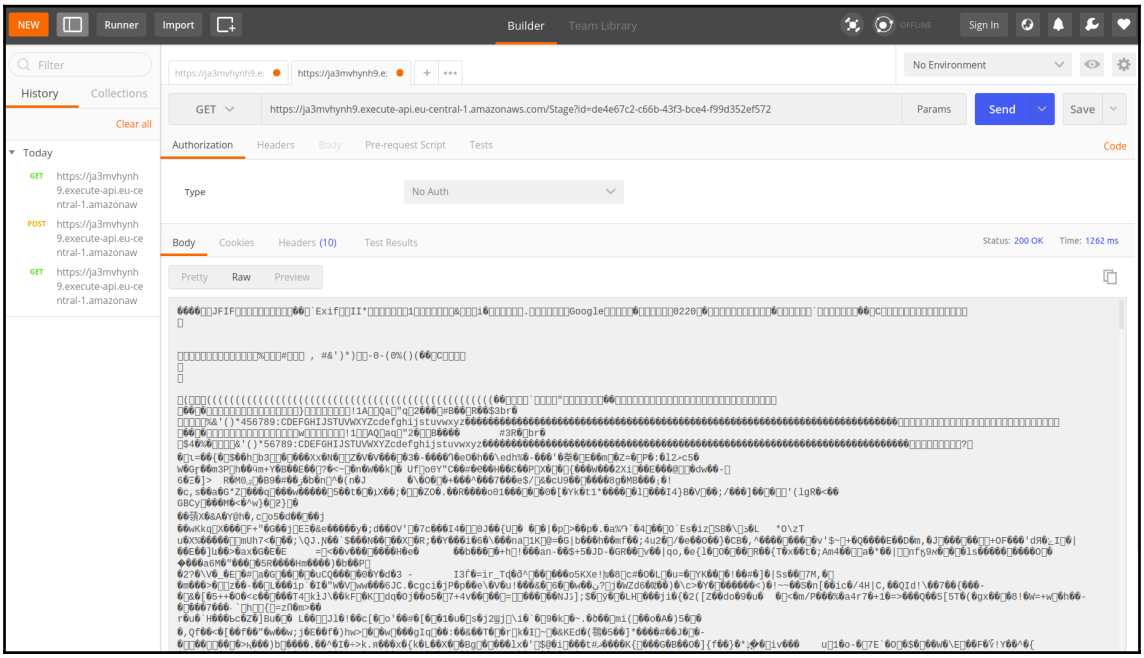

Resources
Actions
/
GET
POST
/ - GET - Method Execution
TEST
Client
Method Request
Auth: NONE
ARN: arn:aws:execute-api:eu-central-1:019859648260:z0c855ixd2/*/GET/
Integration Request
Type: LAMBDA_PROXY
Lambda JavaBlog-devstack-GetArticle-1LI2447WELEQZ
Method Response
Select an integration response.
Integration Response
Proxy integrations cannot be configured to transform responses.

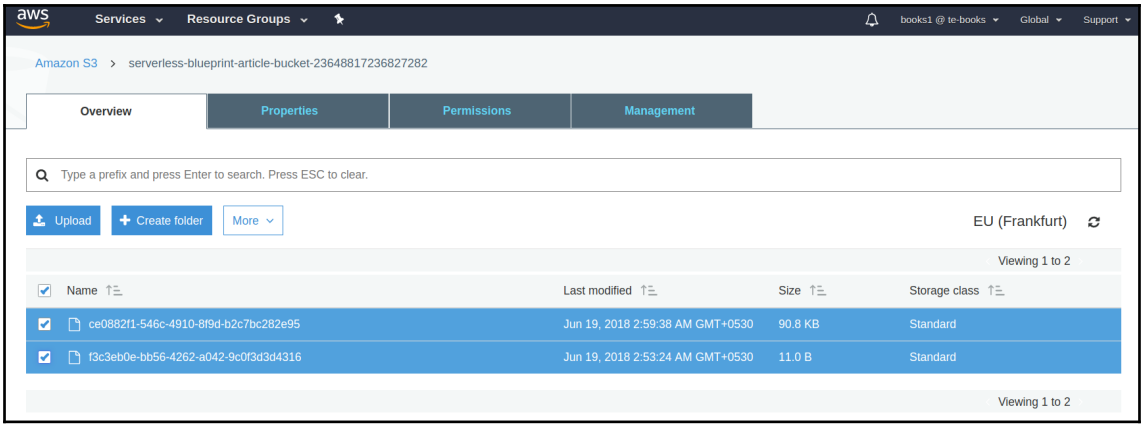

JavaBlog-devstack-GetArticle-1LI2447WELEQZ
Throttle
Qualifiers
Actions
Select a test event..
Test
Save
Configuration
Monitoring
Designer
Add triggers
Click on a trigger from the list below to add it to your function.
API Gateway
AWS IoT
CloudWatch Events
CloudWatch Logs
CodeCommit
Cognito Sync Trigger
DynamoDB
Kinesis
S3
SNS
JavaBlog-devstack-GetArticle-1LI2447WELEQZ
Related functions:
Select a function
API Gateway
Add triggers from the list on the left
AWS Lambda
Amazon CloudWatch
Amazon CloudWatch Logs
Amazon DynamoDB
Amazon DynamoDB Accelerator (DAX)
Amazon EC2
Amazon S3
Function code
Info
Code entry type
Upload a .ZIP or JAR file
Runtime
Java 8
Handler
Info
com.serverless.demo.function.GetArticle
Function package*
Upload
For files larger than 10 MB, consider uploading via S3.
Environment variables
You can define Environment Variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code.
Learn more.
ARTICLE_BUCKET_NAME
serverless-blueprint-article-bucket-23648817236827282
Remove
ARTICLE_TABLE_NAME
serverless-blueprint-article-table
Remove
Key
Value
Remove

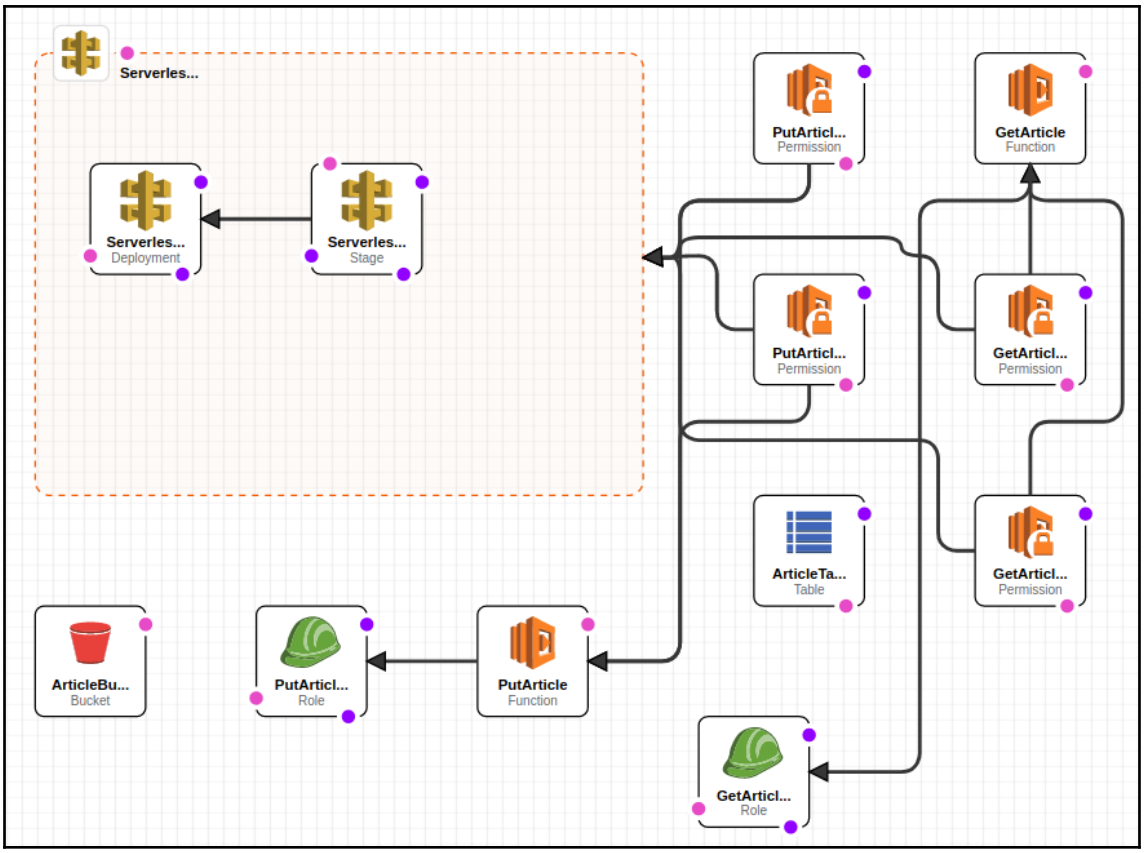

Create table
Delete table
Filter by table name
Name
BlogTable
serverless-blueprint-article-table
serverless-blueprint-article-table
Close
Overview
Items
Metrics
Alarms
Capacity
Indexes
Global Tables
Backups
Triggers
Access control
Tags
Create item
Actions
Scan: [Table] serverless-blueprint-article-table: id
Viewing 0 to 0 items
Scan
[Table] serverless-blueprint-article-table: id
Add filter
Start search
id
An item consists of one or more attributes. Each attribute consists of a name, a data type, and a value. When you read or write an item, the only attributes that are required are those that make up the primary key.
More info


NEW
Runner
Import
Builder
Team Library
Sign In
Filter
History
Collections
Nothing in your history yet. Requests that you send through Postman are automatically saved here.
No Environment
POST
https://ja3mvhynh9.execute-api.eu-central-1.amazonaws.com/Stage/
Params
Send
Save
Authorization
Headers (1)
Body
Pre-request Script
Tests
Code
Key
Value
Description
Bulk Edit
Presets
Content-Type
text/plain
New key
Value
Description
Body
Cookies
Headers (10)
Test Results
Status: 200 OK
Time: 13001 ms
Pretty
Raw
Preview
Successfully inserted article a41d3adc-2150-4967-bca8-a1f572e0695c

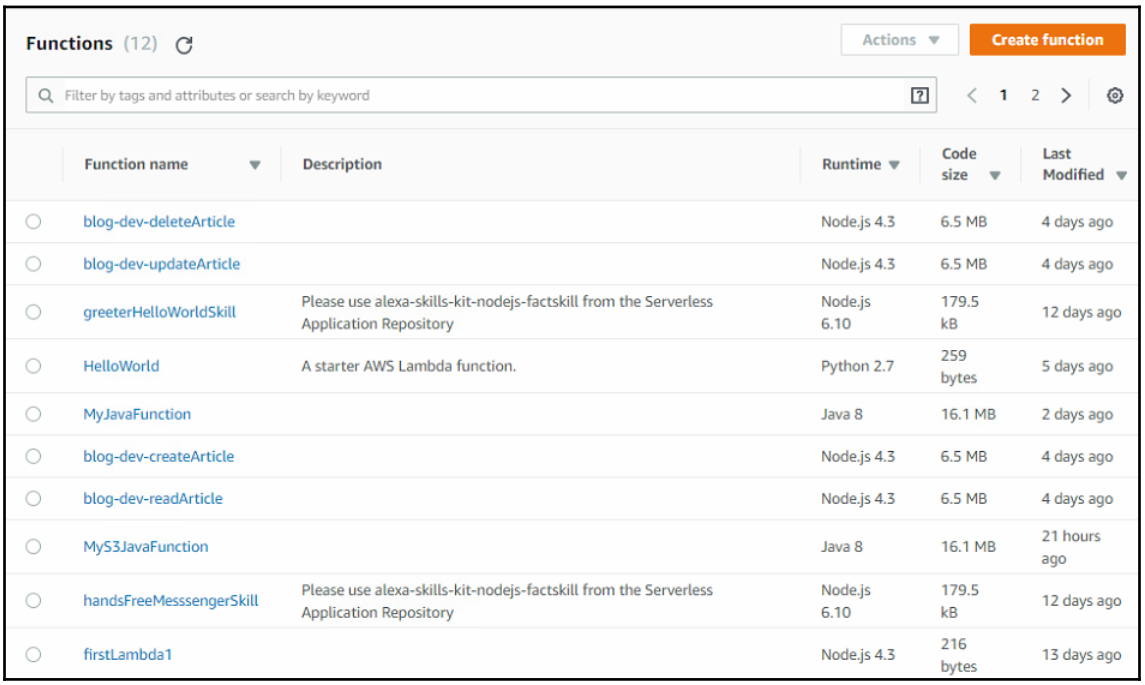

Filter
History
Collections
Clear all
Today
GET
https://ja3mvhynh9.execute-api.eu-central-1.amazonaw
No Environment
GET
https://ja3mvhynh9.execute-api.eu-central-1.amazonaws.com/Stage?id=a41d3adc-2150-4967-bca8-a1f572e0695c
Params
Send
Save
Authorization
Headers
Body
Pre-request Script
Tests
Code
Type
No Auth
Body
Cookies
Headers (10)
Test Results
Status: 200 OK
Time: 10420 ms
Pretty
Raw
Preview
Hello World


NEW
Runner
Import
Builder
Team Library
Sign In
Filter
History
Collections
Clear all
Today
GET
https://ja3mvhynh9.execute-api.eu-central-1.amazonaw
POST
https://ja3mvhynh9.execute-api.eu-central-1.amazonaw
GET
https://ja3mvhynh9.execute-api.eu-central-1.amazonaw
No Environment
GET
https://ja3mvhynh9.execute-api.eu-central-1.amazonaws.com/Stage?id=de4e67c2-c66b-43f3-bce4-f99d352ef572
Params
Send
Save
Authorization
Headers
Body
Pre-request Script
Tests
Code
Type
No Auth
Body
Cookies
Headers (10)
Test Results
Status: 200 OK
Time: 1262 ms
Pretty
Raw
Preview


aws
Services
Resource Groups
books1 @ te-books
Global
Support
Amazon S3 > serverless-blueprint-article-bucket-23648817236827282
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
EU (Frankfurt)
Viewing 1 to 2
Name
Last modified
Size
Storage class
ce0882f1-546c-4910-8f9d-b2c7bc282e95
Jun 19, 2018 2:59:38 AM GMT+0530
90.8 KB
Standard
f3c3eb0e-bb56-4262-a042-9c0f3d3d4316
Jun 19, 2018 2:53:24 AM GMT+0530
11.0 B
Standard
Viewing 1 to 2


Serverles...
Serverles...
Deployment
Serverles...
Stage
PutArticl...
Permission
GetArticle
Function
PutArticl...
Permission
GetArticl...
Permission
ArticleTa...
Table
GetArticl...
Permission
ArticleBu...
Bucket
PutArticl...
Role
PutArticle
Function
GetArticl...
Role

Chapter 5:
Programming AWS Lambda with Python



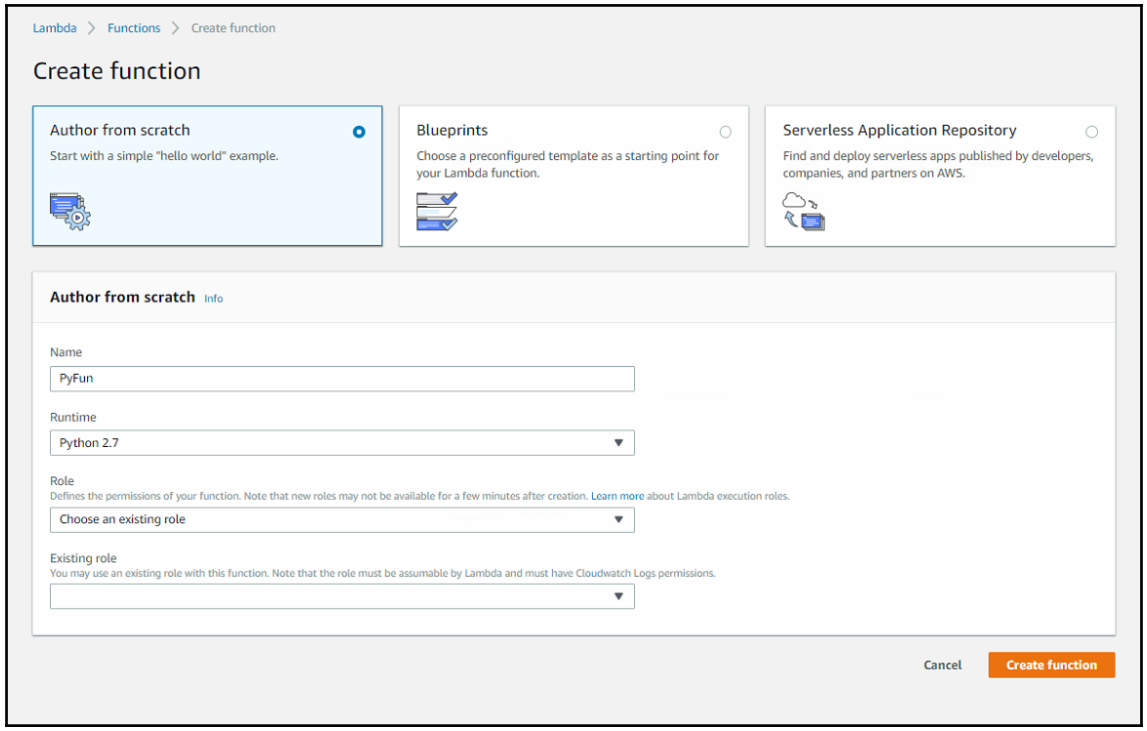
Functions (12)

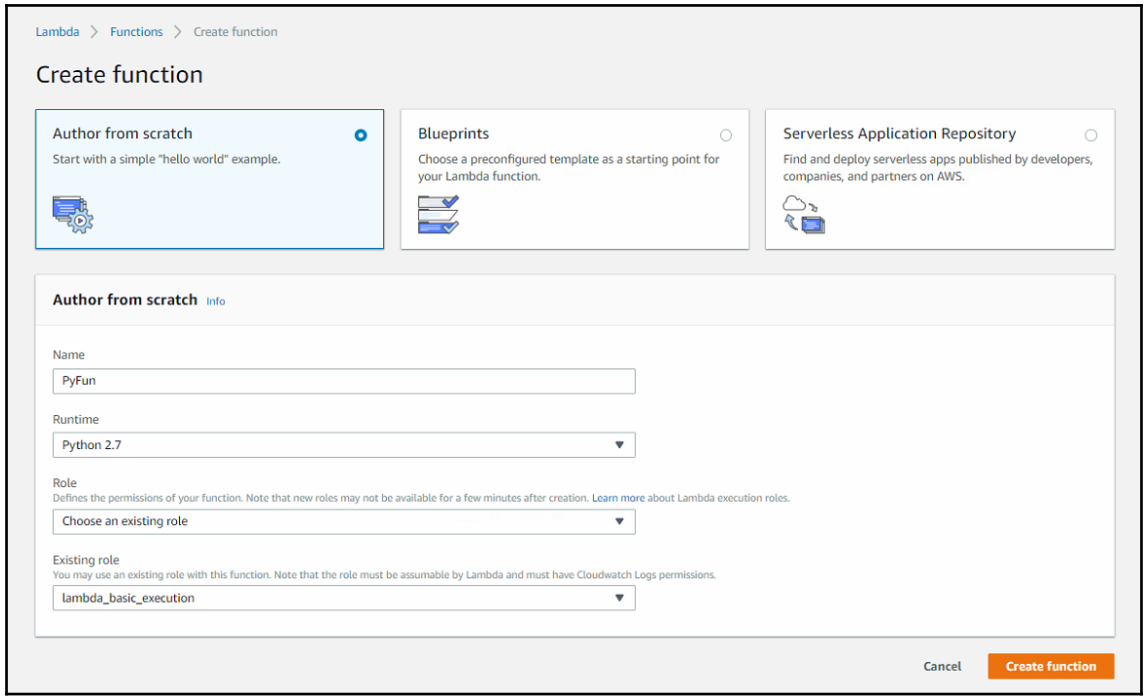
Actions | Create function

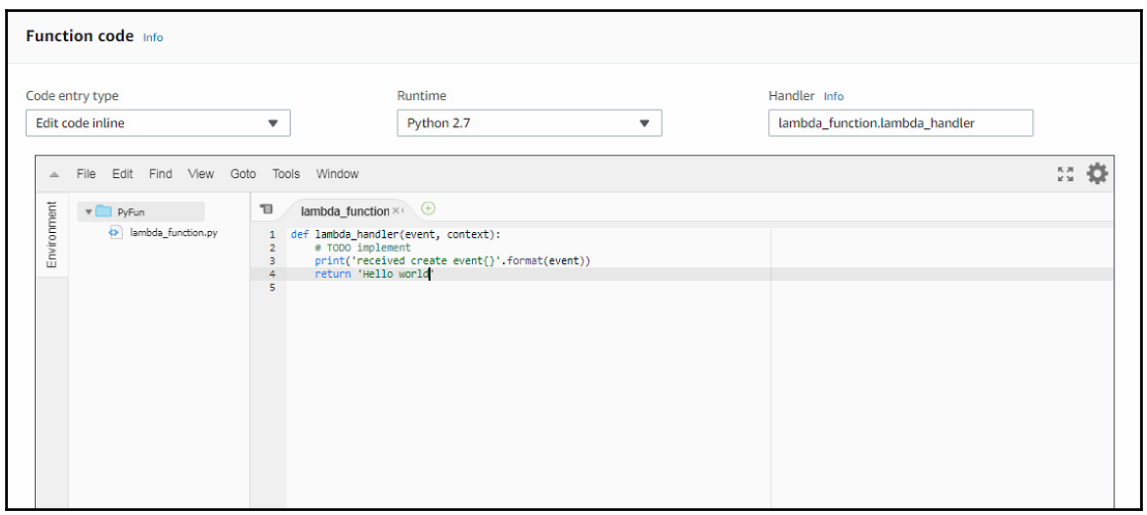
Filter by tags and attributes or search by keyword

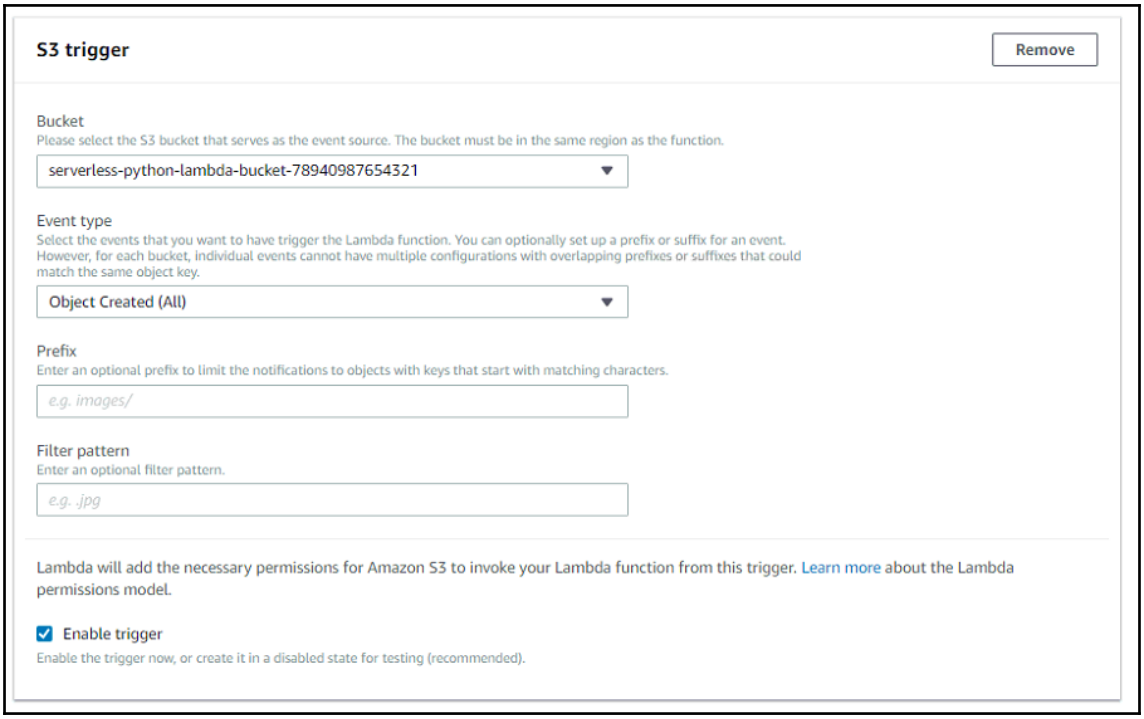
1 2

Function name	Description	Runtime	Code size	Last Modified
blog-dev-deleteArticle		Node.js 4.3	6.5 MB	4 days ago
blog-dev-updateArticle		Node.js 4.3	6.5 MB	4 days ago
greeterHelloWorldSkill	Please use alexa-skills-kit-nodejs-factskill from the Serverless Application Repository	Node.js 6.10	179.5 kB	12 days ago
HelloWorld	A starter AWS Lambda function.	Python 2.7	259 bytes	5 days ago
MyJavaFunction		Java 8	16.1 MB	2 days ago
blog-dev-createArticle		Node.js 4.3	6.5 MB	4 days ago
blog-dev-readArticle		Node.js 4.3	6.5 MB	4 days ago
MyS3JavaFunction		Java 8	16.1 MB	21 hours ago
handsFreeMesssengerSkill	Please use alexa-skills-kit-nodejs-factskill from the Serverless Application Repository	Node.js 6.10	179.5 kB	12 days ago
firstLambda1		Node.js 4.3	216 bytes	13 days ago


Lambda
Functions
Create function
Create function
Author from scratch
Start with a simple "hello world" example.
Blueprints
Choose a preconfigured template as a starting point for your Lambda function.
Serverless Application Repository
Find and deploy serverless apps published by developers, companies, and partners on AWS.
Author from scratch
Info
Name
PyFun
Runtime
Python 2.7
Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. Learn more about Lambda execution roles.
Choose an existing role
Existing role
You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.
Cancel
Create function


Lambda > Functions > Create function
Create function
Author from scratch
Start with a simple "hello world" example.
Blueprints
Choose a preconfigured template as a starting point for your Lambda function.
Serverless Application Repository
Find and deploy serverless apps published by developers, companies, and partners on AWS.
Author from scratch Info
Name
PyFun
Runtime
Python 2.7
Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. Learn more about Lambda execution roles.
Choose an existing role
Existing role
You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.
lambda_basic_execution
Cancel
Create function


Function code Info
Code entry type
Edit code inline
Runtime
Python 2.7
Handler Info
lambda_function.lambda_handler
File Edit Find View Goto Tools Window
Environment
PyFun
lambda_function.py
lambda_function
def lambda_handler(event, context):
# TODO implement
print('received create event{}'.format(event))
return 'Hello world


S3 trigger
Remove
Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.
serverless-python-lambda-bucket-78940987654321
Event type
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
Object Created (All)
Prefix
Enter an optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/
Filter pattern
Enter an optional filter pattern.
e.g. .jpg
Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.
Enable trigger
Enable the trigger now, or create it in a disabled state for testing (recommended).

Lambda function code

Code is pre-configured by the chosen blueprint. You can configure it after you create the function.

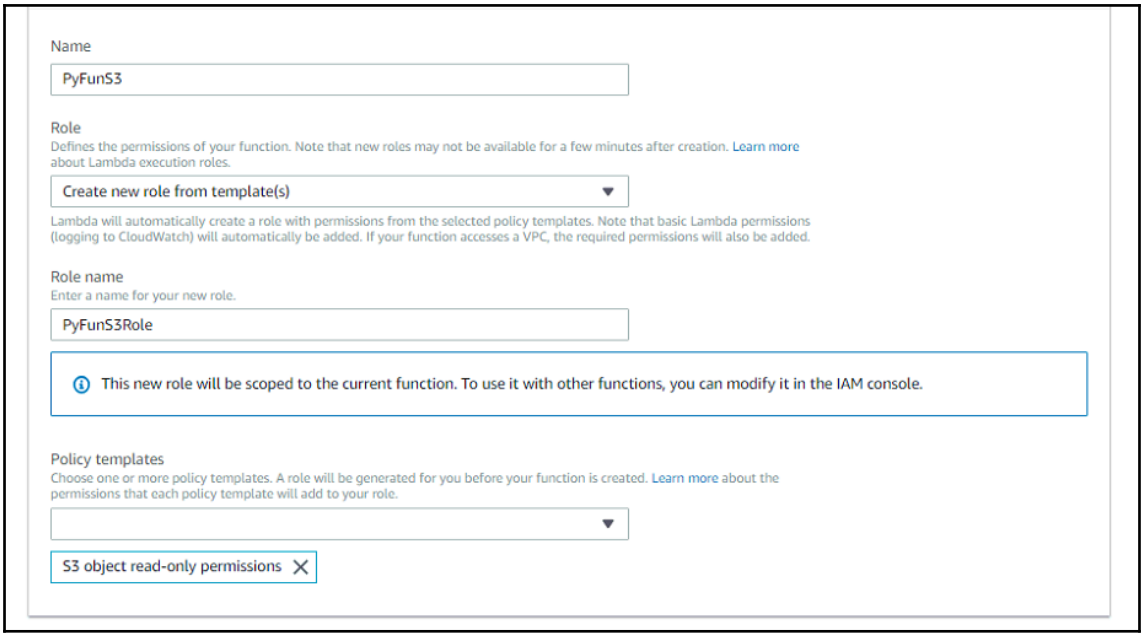
Runtime
Python 2.7

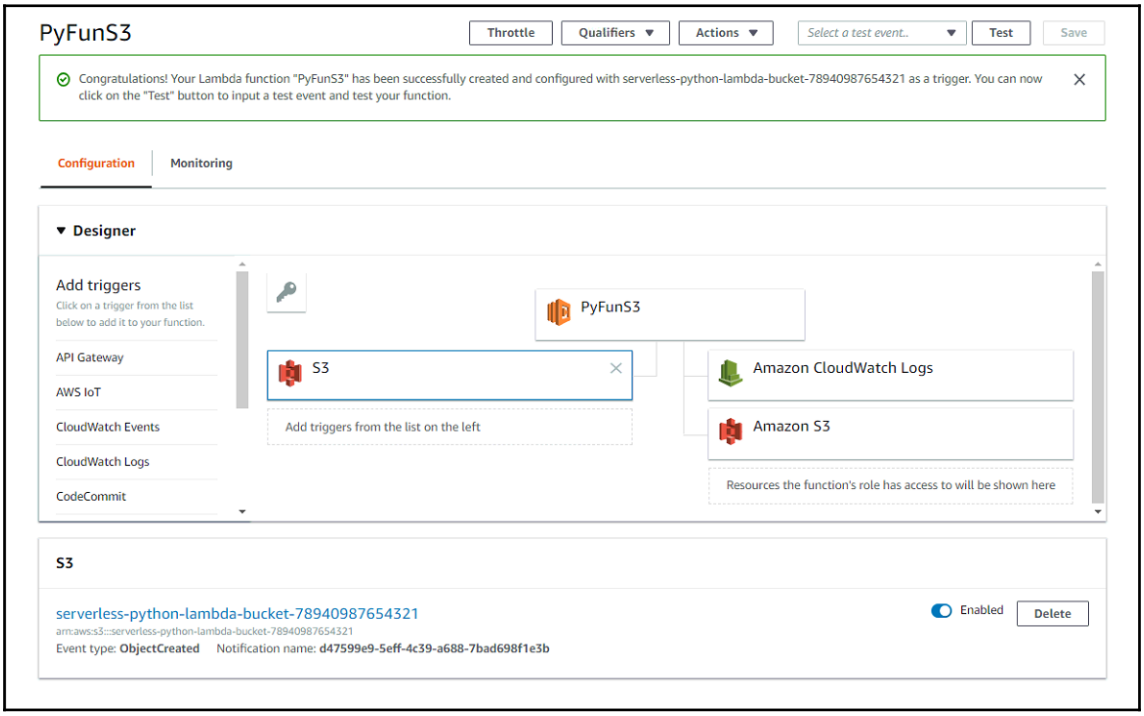
```
1  from __future__ import print_function
2
3  import json
4  import urllib
5  import boto3
6
7  print('Loading function')
8
9  s3 = boto3.client('s3')
10
11
12 def lambda_handler(event, context):
13     #print("Received event: " + json.dumps(event, indent=2))
14
15     # Get the object from the event and show its content type
16     bucket = event['Records'][0]['s3']['bucket']['name']
17     key = urllib.unquote_plus(event['Records'][0]['s3']['object']['key'].encode('utf8'))
18     try:
19         response = s3.get_object(Bucket=bucket, Key=key)
20         print("CONTENT TYPE: " + response['ContentType'])
21         return response['ContentType']
22     except Exception as e:
23         print(e)
24         print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region a
25         raise e
26
```

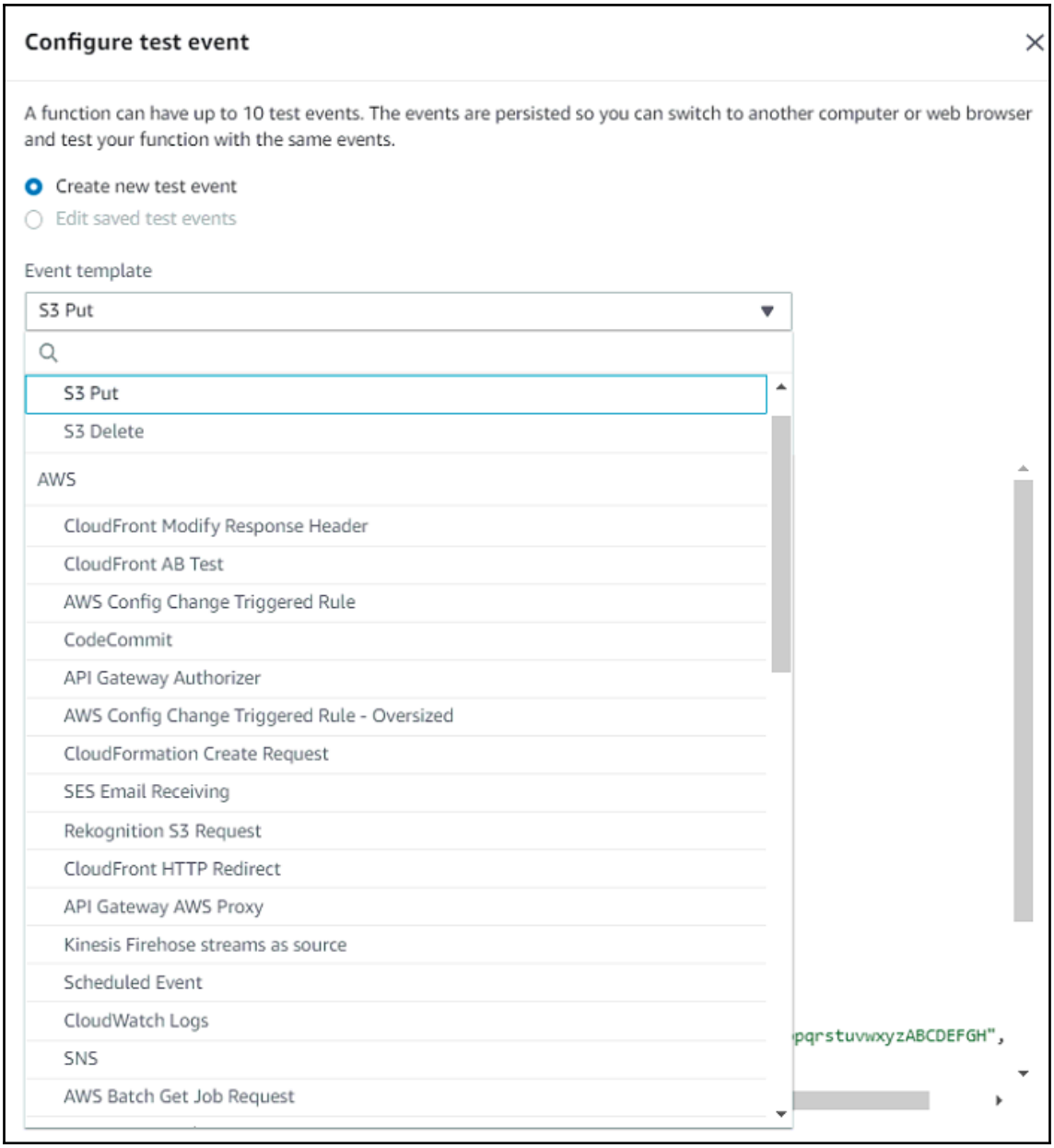
```
s3 = boto3.client('s3')
```

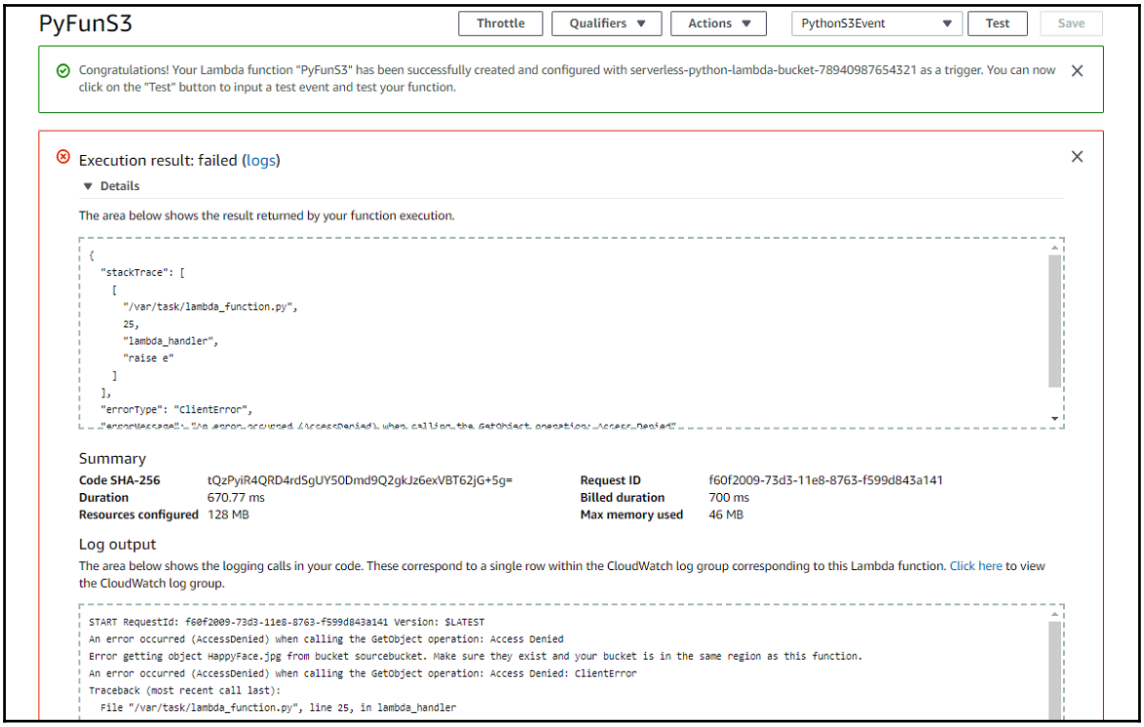
```
bucket = event['Records'][0]['s3']['bucket']['name']
key = urllib.unquote_plus(event['Records'][0]['s3']['object']['key'].encode('utf8'))
```

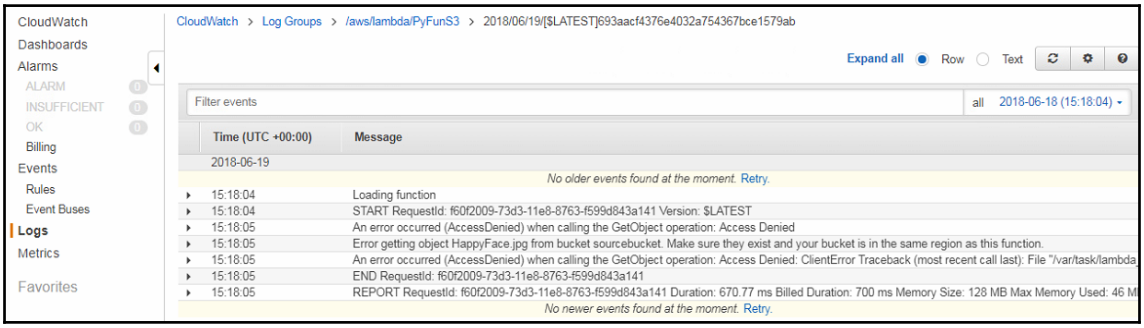
```
    response = s3.get_object(Bucket=bucket, Key=key)
    print("CONTENT TYPE: " + response['ContentType'])
    return response['ContentType']
except Exception as e:
    print(e)
    print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in
    raise e
```

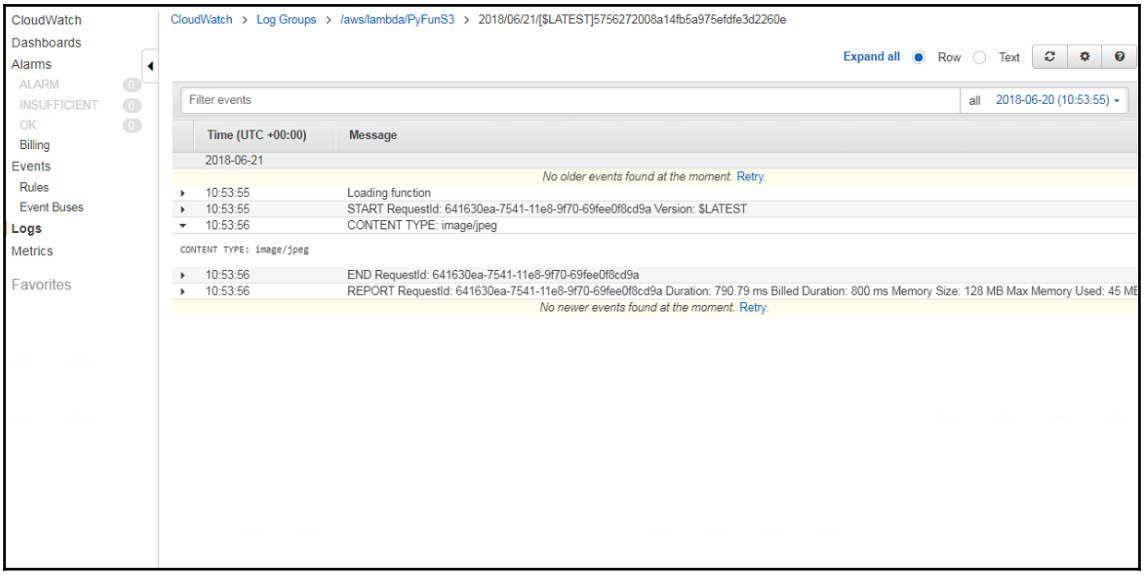

Name
PyFunS3
Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. Learn more about Lambda execution roles.
Create new role from template(s)
Lambda will automatically create a role with permissions from the selected policy templates. Note that basic Lambda permissions (logging to CloudWatch) will automatically be added. If your function accesses a VPC, the required permissions will also be added.
Role name
Enter a name for your new role.
PyFunS3Role
This new role will be scoped to the current function. To use it with other functions, you can modify it in the IAM console.
Policy templates
Choose one or more policy templates. A role will be generated for you before your function is created. Learn more about the permissions that each policy template will add to your role.
S3 object read-only permissions


PyFunS3
Throttle
Qualifiers
Actions
Select a test event..
Test
Save
Congratulations! Your Lambda function "PyFunS3" has been successfully created and configured with serverless-python-lambda-bucket-78940987654321 as a trigger. You can now click on the "Test" button to input a test event and test your function.
Configuration
Monitoring
Designer
Add triggers
Click on a trigger from the list below to add it to your function.
API Gateway
AWS IoT
CloudWatch Events
CloudWatch Logs
CodeCommit
PyFunS3
S3
Add triggers from the list on the left
Amazon CloudWatch Logs
Amazon S3
Resources the function's role has access to will be shown here
S3
serverless-python-lambda-bucket-78940987654321
arn:aws:s3:::serverless-python-lambda-bucket-78940987654321
Event type: ObjectCreated
Notification name: d47599e9-5eff-4c39-a688-7bad698f1e3b
Enabled
Delete


Configure test event
A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.
Create new test event
Edit saved test events
Event template
S3 Put
S3 Put
S3 Delete
AWS
CloudFront Modify Response Header
CloudFront AB Test
AWS Config Change Triggered Rule
CodeCommit
API Gateway Authorizer
AWS Config Change Triggered Rule - Oversized
CloudFormation Create Request
SES Email Receiving
Rekognition S3 Request
CloudFront HTTP Redirect
API Gateway AWS Proxy
Kinesis Firehose streams as source
Scheduled Event
CloudWatch Logs
SNS
AWS Batch Get Job Request
pqrstuvwxyzABCDEFGH",


PyFunS3
Throttle
Qualifiers
Actions
PythonS3Event
Test
Save
Congratulations! Your Lambda function "PyFunS3" has been successfully created and configured with serverless-python-lambda-bucket-78940987654321 as a trigger. You can now click on the "Test" button to input a test event and test your function.
Execution result: failed (logs)
Details
The area below shows the result returned by your function execution.
{
"stackTrace": [
[
"/var/task/lambda_function.py",
25,
"lambda_handler",
"raise e"
]
],
"errorType": "ClientError",
Summary
Code SHA-256 tQzPyiR4QRD4rdSgUY50Dmd9Q2gkJz6exVBT62jG+5g=
Request ID f60f2009-73d3-11e8-8763-f599d843a141
Duration 670.77 ms
Billed duration 700 ms
Resources configured 128 MB
Max memory used 46 MB
Log output
The area 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: f60f2009-73d3-11e8-8763-f599d843a141 Version: $LATEST
An error occurred (AccessDenied) when calling the GetObject operation: Access Denied
Error getting object HappyFace.jpg from bucket sourcebucket. Make sure they exist and your bucket is in the same region as this function.
An error occurred (AccessDenied) when calling the GetObject operation: Access Denied: ClientError
Traceback (most recent call last):
File "/var/task/lambda_function.py", line 25, in lambda_handler


CloudWatch
Dashboards
Alarms
ALARM
INSUFFICIENT
OK
Billing
Events
Rules
Event Buses
Logs
Metrics
Favorites
CloudWatch > Log Groups > /aws/lambda/PyFunS3 > 2018/06/19/[$LATEST]693aacf4376e4032a754367bce1579ab
Expand all
Row
Text
Filter events
all
2018-06-18 (15:18:04)
Time (UTC +00:00)
Message
2018-06-19
No older events found at the moment. Retry.
15:18:04 Loading function
15:18:04 START RequestId: f60f2009-73d3-11e8-8763-f599d843a141 Version: $LATEST
15:18:05 An error occurred (AccessDenied) when calling the GetObject operation: Access Denied
15:18:05 Error getting object HappyFace.jpg from bucket sourcebucket. Make sure they exist and your bucket is in the same region as this function.
15:18:05 An error occurred (AccessDenied) when calling the GetObject operation: Access Denied: ClientError Traceback (most recent call last): File "/var/task/lambda
15:18:05 END RequestId: f60f2009-73d3-11e8-8763-f599d843a141
15:18:05 REPORT RequestId: f60f2009-73d3-11e8-8763-f599d843a141 Duration: 670.77 ms Billed Duration: 700 ms Memory Size: 128 MB Max Memory Used: 46 M
No newer events found at the moment. Retry.



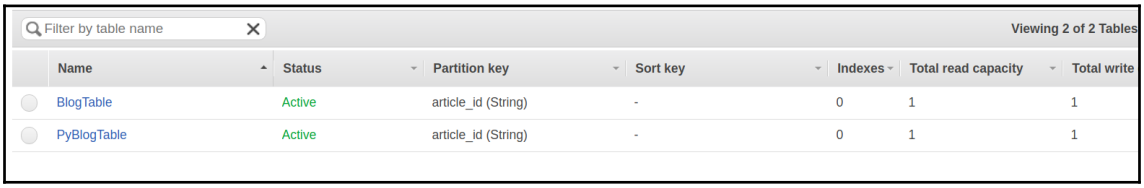

```
Serverless: Successfully generated boilerplate for template: "aws-python"
Admin@Admin:~/Desktop/admin/serverless/programming-aws-lambda-master/python$ sls invoke local -f hello
{
    "body": "{\"input\": {}, \"message\": \"Go Serverless v1.0! Your function executed successfully!\"}",
    "statusCode": 200
}
```

```
Admin@Admin:~/Desktop/admin/serverless/programming-aws-lambda-master/python$ 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 .zip file to S3 (12.49 KB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
..............
Serverless: Stack update finished...
Service Information
service: pyblog
stage: dev
region: eu-central-1
stack: pyblog-dev
api keys:
  None
endpoints:
  None
functions:
  hello: pyblog-dev-hello
```

```
functions:
  hello: pyblog-dev-hello
Admin@Admin:~/Desktop/admin/serverless/programming-aws-lambda-master/python$ sls invoke -f hello
{
    "body": "{\"input\": {}, \"message\": \"Go Serverless v1.0! Your function executed successfully!\"}",
    "statusCode": 200
}
```

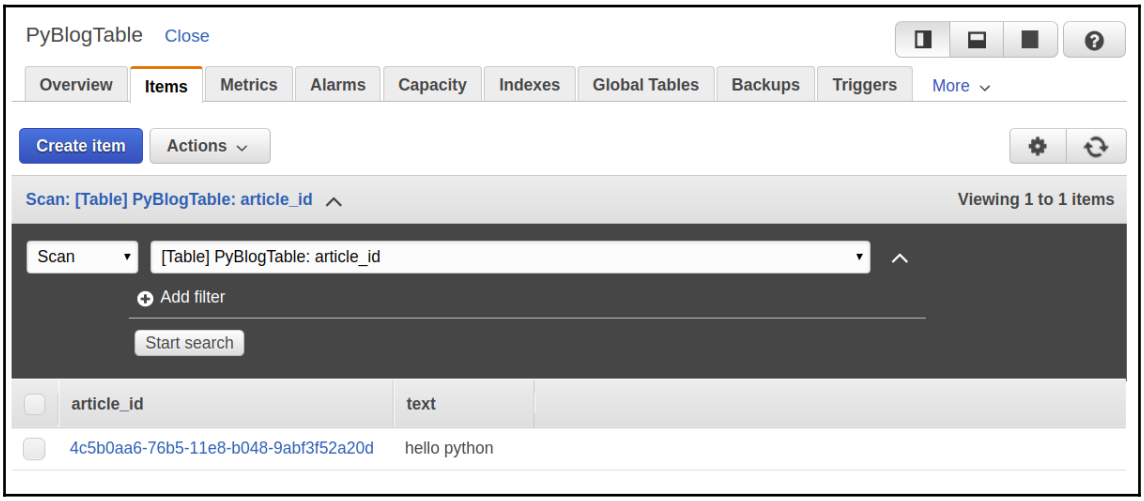
```
Admin@Admin:~/Desktop/admin/serverless/programming-aws-lambda-master/python$ sls deploy
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service .zip file to S3 (12.52 KB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.....................................
Serverless: Stack update finished...
Service Information
service: pyblog
stage: dev
region: eu-central-1
stack: pyblog-dev
api keys:
  None
endpoints:
  POST - https://jppj97nlo7.execute-api.eu-central-1.amazonaws.com/dev/create
functions:
  create: pyblog-dev-create
```

```
functions:
  create: pyblog-dev-create
Admin@Admin:~/Desktop/admin/serverless/programming-aws-lambda-master/python$ sls invoke -f create
{
    "body": "{\"message\": \"Created new article\"}",
    "statusCode": 200
}
```



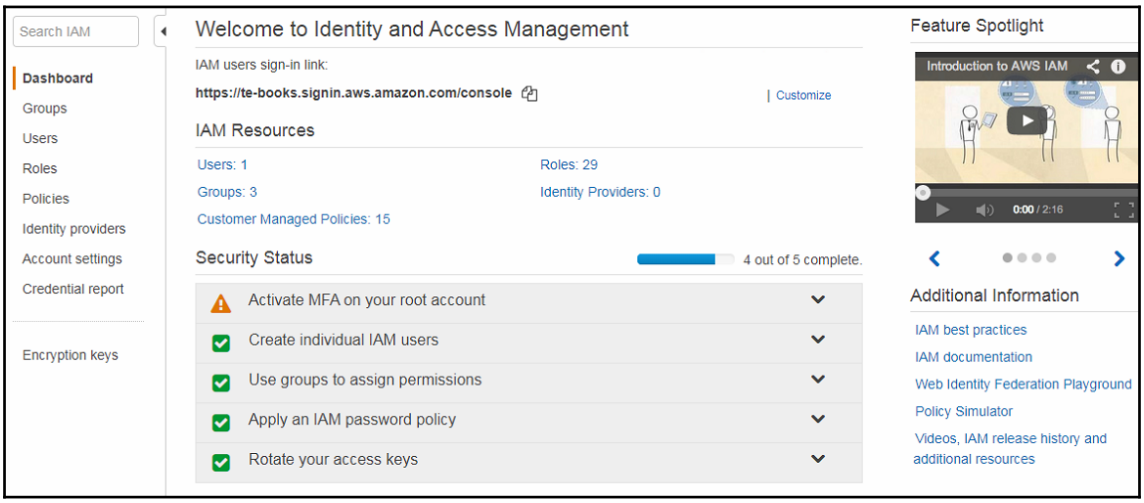
```
Admin@Admin:~/Desktop/admin/serverless/programming-aws-lambda-master/python$ sls deploy
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service .zip file to S3 (12.69 KB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.................
Serverless: Stack update finished...
Service Information
service: pyblog
stage: dev
region: eu-central-1
stack: pyblog-dev
api keys:
  None
endpoints:
  POST - https://jppj97nlo7.execute-api.eu-central-1.amazonaws.com/dev/create
functions:
  create: pyblog-dev-create
```

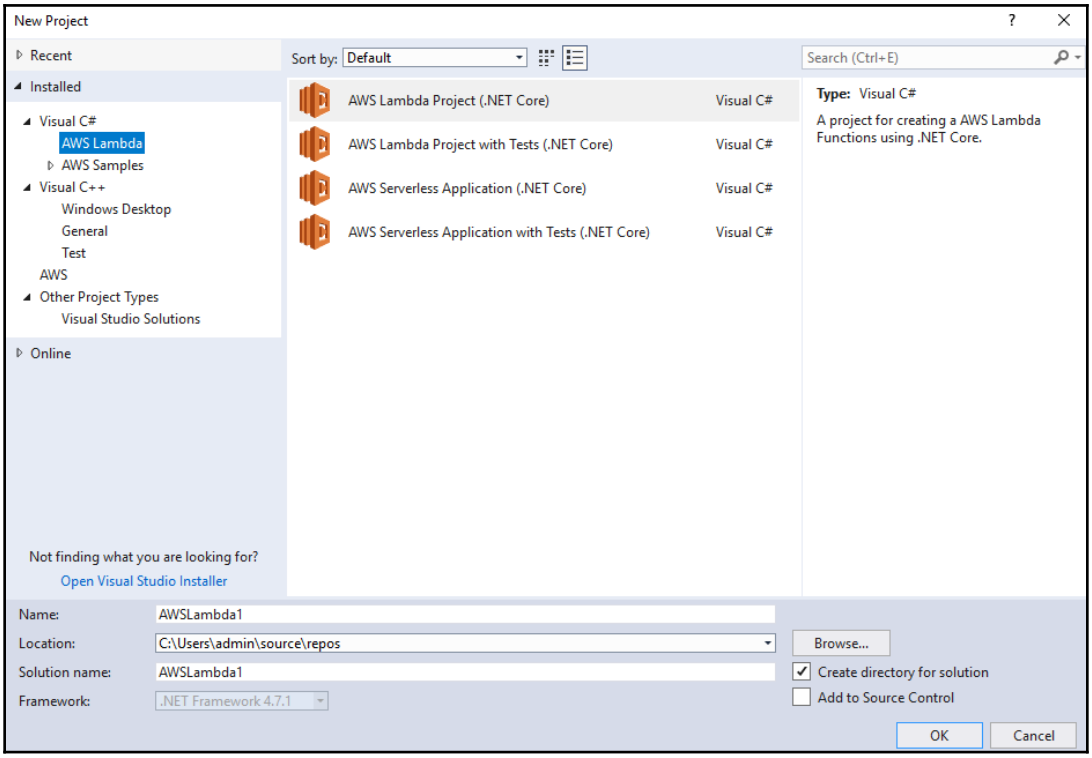
```
Admin@Admin:~/Desktop/admin/serverless/programming-aws-lambda-master/python$ sls invoke -f create -p event.json
{
    "body": "{\"ResponseMetadata\": {\"RetryAttempts\": 0, \"HTTPStatusCode\": 200, \"RequestId\": \"742VSAUSNUQG0JFELD073VHM4RVV4KQNSO5AEMVJF66Q9ASUAAJG\", \"HTTPHeaders\": {\"x-amzn-requestid\": \"742VSAUSNUQG0JFELD073VHM4RVV4KQNSO5AEMVJF66Q9ASUAAJG\", \"content-length\": \"2\", \"server\": \"Server\", \"connection\": \"keep-alive\", \"x-amz-crc32\": \"2745614147\", \"date\": \"Sat, 23 Jun 2018 07:16:07 GMT\", \"content-type\": \"application/x-amz-json-1.0\"}}}",
    "statusCode": 200
}
```

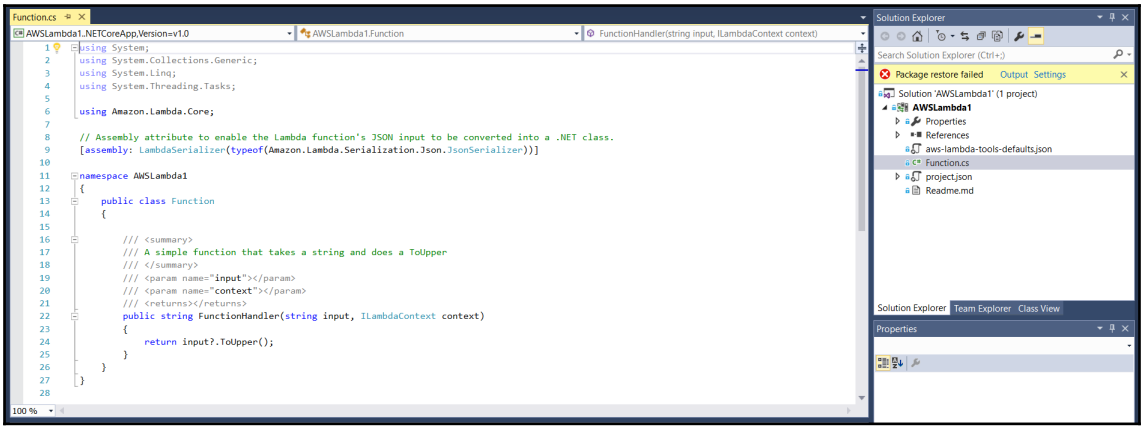


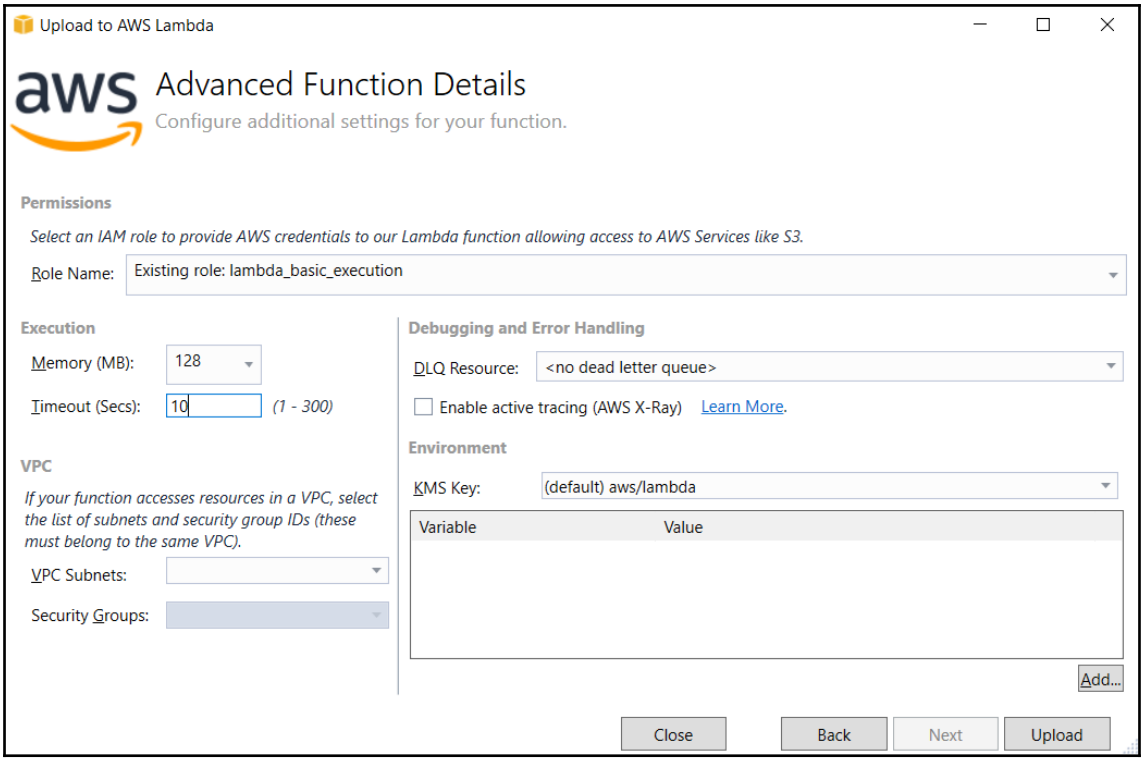
```
Admin@Admin:~/Desktop/admin/serverless/programming-aws-lambda-master/python$ sls deploy
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service .zip file to S3 (14.27 KB)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
.....................................................................
Serverless: Stack update finished...
Service Information
service: pyblog
stage: dev
region: eu-central-1
stack: pyblog-dev
api keys:
  None
endpoints:
  POST - https://jppj97nlo7.execute-api.eu-central-1.amazonaws.com/dev/articles
  GET - https://jppj97nlo7.execute-api.eu-central-1.amazonaws.com/dev/articles/{id}
  PUT - https://jppj97nlo7.execute-api.eu-central-1.amazonaws.com/dev/articles
  DELETE - https://jppj97nlo7.execute-api.eu-central-1.amazonaws.com/dev/articles
functions:
  create: pyblog-dev-create
  read: pyblog-dev-read
  update: pyblog-dev-update
  delete: pyblog-dev-delete
```

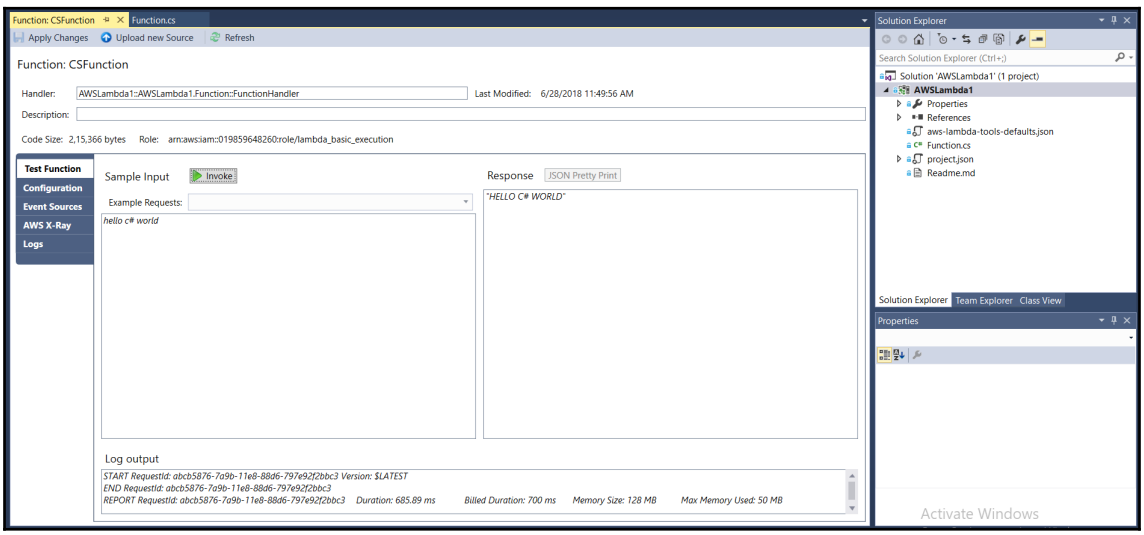
Chapter 6:
Programming AWS Lambda with C#




New Project
Recent
Installed
Visual C#
AWS Lambda
AWS Samples
Visual C++
Windows Desktop
General
Test
AWS
Other Project Types
Visual Studio Solutions
Online
Not finding what you are looking for?
Open Visual Studio Installer
Sort by: Default
Search (Ctrl+E)
AWS Lambda Project (.NET Core) Visual C#
AWS Lambda Project with Tests (.NET Core) Visual C#
AWS Serverless Application (.NET Core) Visual C#
AWS Serverless Application with Tests (.NET Core) Visual C#
Type: Visual C#
A project for creating a AWS Lambda Functions using .NET Core.
Name: AWSLambda1
Location: C:\Users\admin\source\repos
Solution name: AWSLambda1
Framework: .NET Framework 4.7.1
Browse...
Create directory for solution
Add to Source Control
OK
Cancel


Function.cs
AWSLambda1..NETCoreApp,Version=v1.0
AWSLambda1.Function
FunctionHandler(string input, ILambdaContext context)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Amazon.Lambda.Core;
// Assembly attribute to enable the Lambda function's JSON input to be converted into a .NET class.
[assembly: LambdaSerializer(typeof(Amazon.Lambda.Serialization.Json.JsonSerializer))]
namespace AWSLambda1
{
public class Function
{
/// <summary>
/// A simple function that takes a string and does a ToUpper
/// </summary>
/// <param name="input"></param>
/// <param name="context"></param>
/// <returns></returns>
public string FunctionHandler(string input, ILambdaContext context)
{
return input?.ToUpper();
}
}
}
100 %
Solution Explorer
Search Solution Explorer (Ctrl+;)
Package restore failed
Output
Settings
Solution 'AWSLambda1' (1 project)
AWSLambda1
Properties
References
aws-lambda-tools-defaults.json
Function.cs
project.json
Readme.md
Solution Explorer
Team Explorer
Class View
Properties


Upload to AWS Lambda
Advanced Function Details
Configure additional settings for your function.
Permissions
Select an IAM role to provide AWS credentials to our Lambda function allowing access to AWS Services like S3.
Role Name:
Existing role: lambda_basic_execution
Execution
Memory (MB):
128
Timeout (Secs):
10
(1 - 300)
VPC
If your function accesses resources in a VPC, select the list of subnets and security group IDs (these must belong to the same VPC).
VPC Subnets:
Security Groups:
Debugging and Error Handling
DLQ Resource:
<no dead letter queue>
Enable active tracing (AWS X-Ray)
Learn More.
Environment
KMS Key:
(default) aws/lambda
Variable
Value
Add...
Close
Back
Next
Upload


Function: CSFunction
Function.cs
Apply Changes
Upload new Source
Refresh
Function: CSFunction
Handler:
AWSLambda1::AWSLambda1.Function::FunctionHandler
Last Modified:
6/28/2018 11:49:56 AM
Description:
Code Size:
2,15,366 bytes
Role:
arn:aws:iam::019859648260:role/lambda_basic_execution
Test Function
Configuration
Event Sources
AWS X-Ray
Logs
Sample Input
Invoke
Example Requests:
hello c# world
Response
JSON Pretty Print
"HELLO C# WORLD"
Log output
START RequestId: abcb5876-7a9b-11e8-88d6-797e92f2bbc3 Version: $LATEST
END RequestId: abcb5876-7a9b-11e8-88d6-797e92f2bbc3
REPORT RequestId: abcb5876-7a9b-11e8-88d6-797e92f2bbc3 Duration: 685.89 ms
Billed Duration: 700 ms
Memory Size: 128 MB
Max Memory Used: 50 MB
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'AWSLambda1' (1 project)
AWSLambda1
Properties
References
aws-lambda-tools-defaults.json
Function.cs
project.json
Readme.md
Solution Explorer
Team Explorer
Class View
Properties
Activate Windows

Function name	Description	Runtime	Code size	Last Modified
cs-serverless-stack-AddBlog-16BU9MMEDDA8S	Function to get add a blog	C# (.NET Core 1.0)	577.8 kB	in 5 hours
cs-serverless-stack-GetBlogs-MLXIN1Z5SA1E	Function to get a list of blogs	C# (.NET Core 1.0)	577.8 kB	in 5 hours
cs-serverless-stack-GetBlog-F0RJYMRGX2ZX	Function to get a single blog	C# (.NET Core 1.0)	577.8 kB	in 5 hours
cs-serverless-stack-RemoveBlog-IASTKIJXJH01	Function to remove a blog	C# (.NET Core 1.0)	577.8 kB	in 5 hours

CsBlogTable Close

Overview | Items | Metrics | Alarms | Capacity | Indexes | Global Tables | Backups | Triggers | Access control | Tags

Recent alerts

No CloudWatch alarms have been triggered for this table.

Stream details

Stream enabled	No
View type	-
Latest stream ARN	-

Manage Stream

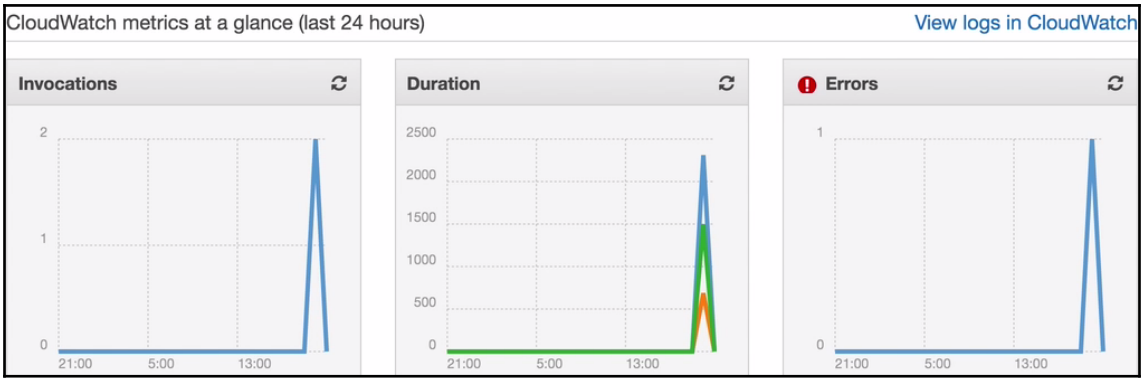
Table details

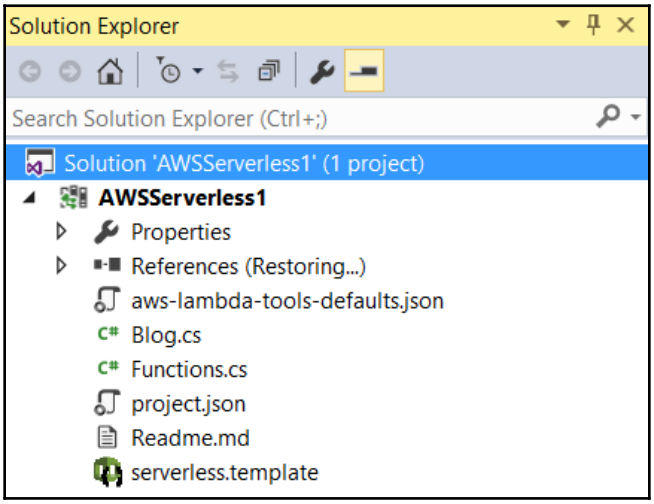
Table name	CsBlogTable
Primary partition key	Id (String)
Primary sort key	-
Point-in-time recovery	DISABLED Enable
Encryption	DISABLED
Time to live attribute	DISABLED Manage TTL
Table status	Active
Creation date	June 28, 2018 at 1:31:12 PM UTC+5:30
Provisioned read capacity units	1 (Auto Scaling Disabled)
Provisioned write capacity units	1 (Auto Scaling Disabled)
Last decrease time	-
Last increase time	-

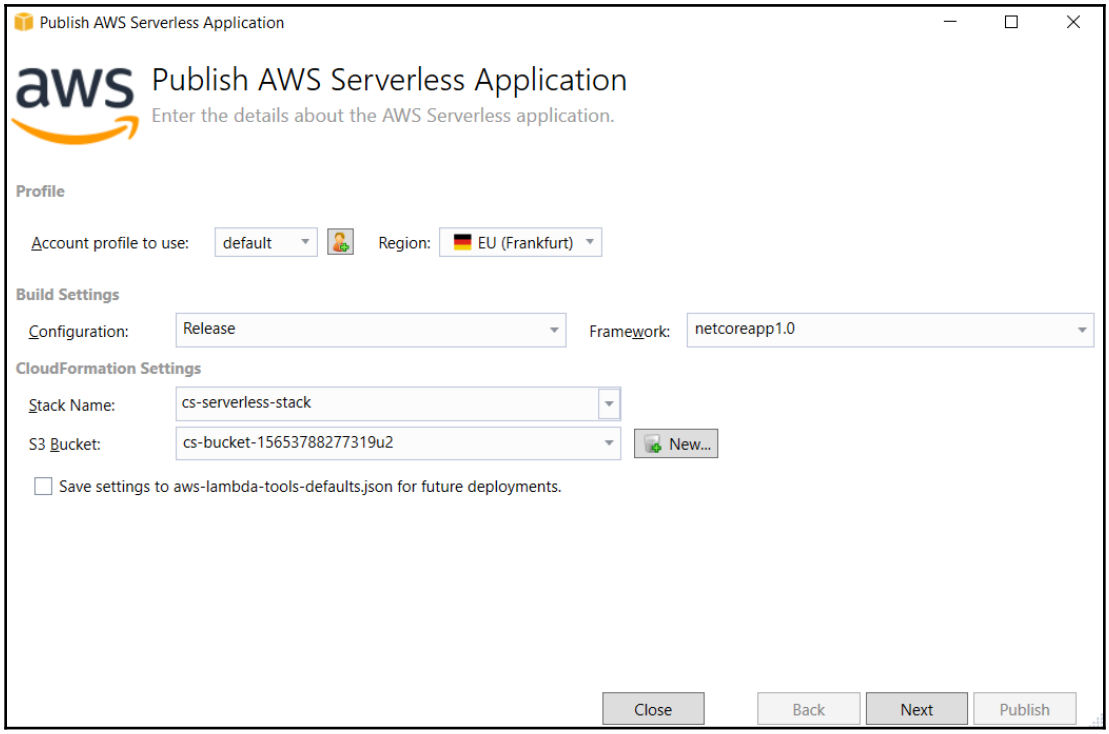
Activate Windows
Go to Settings to activate Windows.

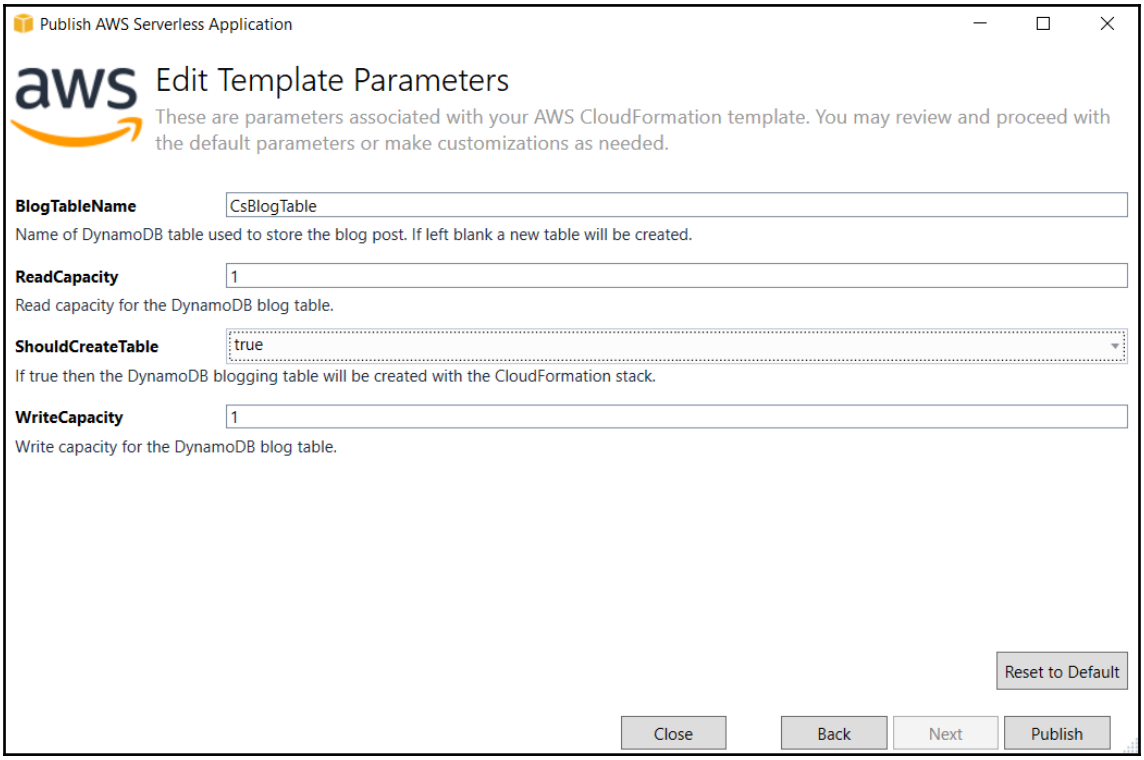
Filter

Function name	Description	Runtime	Code size	Last Modified
CSFunction		C#	209.8 kB	2 minutes ago


CloudWatch metrics at a glance (last 24 hours)
View logs in CloudWatch
Invocations
Duration
Errors
2
1
0
2500
2000
1500
1000
500
0
1
0
21:00
5:00
13:00


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'AWSServerless1' (1 project)
AWSServerless1
Properties
References (Restoring...)
aws-lambda-tools-defaults.json
Blog.cs
Functions.cs
project.json
Readme.md
serverless.template


Publish AWS Serverless Application
Publish AWS Serverless Application
Enter the details about the AWS Serverless application.
Profile
Account profile to use:
default
Region:
EU (Frankfurt)
Build Settings
Configuration:
Release
Framework:
netcoreapp1.0
CloudFormation Settings
Stack Name:
cs-serverless-stack
S3 Bucket:
cs-bucket-15653788277319u2
New...
Save settings to aws-lambda-tools-defaults.json for future deployments.
Close
Back
Next
Publish


Publish AWS Serverless Application
Edit Template Parameters
These are parameters associated with your AWS CloudFormation template. You may review and proceed with the default parameters or make customizations as needed.
BlogTableName
CsBlogTable
Name of DynamoDB table used to store the blog post. If left blank a new table will be created.
ReadCapacity
1
Read capacity for the DynamoDB blog table.
ShouldCreateTable
true
If true then the DynamoDB blogging table will be created with the CloudFormation stack.
WriteCapacity
1
Write capacity for the DynamoDB blog table.
Reset to Default
Close
Back
Next
Publish