

Chapter 1: Amazon Web Services for Serverless

The screenshot shows the AWS Lambda Management console homepage. At the top, there's a navigation bar with links for Apps, SPA, VIM, ZooKeeper, Backbone MVC, Django RealTime, Heroku, Git, Elastic Search, Qura, SPA, VIM, and Other bookmarks. The user is signed in as Abdul Wahid from N. Virginia. Below the navigation, there's a breadcrumb trail showing the path: Lambda Management > Lambda > Home. The main content area has a dark header "COMPUTE" and a sub-header "AWS Lambda". The main text reads: "lets you run code without thinking about servers." A note below states: "You pay only for the compute time you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration." To the right, there's a "Get started" section with a button labeled "Create a function". Below this, there's a "How it works" section with a note about triggering code from other services and endpoints, and a link to "Read more in FAQs". On the right side, there's a "More resources" sidebar with links to Documentation, API reference, and Serverless Application Model (SAM). At the bottom, there are links for Feedback, English (US), and various legal notices.

AWS Lambda Functions > Create function

Create function

Author from scratch **Blueprints** Serverless Application Repository

Blueprints Info Export

Add filter keyword : hello-world

Blueprints

- greengrass-hello-world
- hello-world
- hello-world-python
- hello-world-python3
- greengrass-hello-world-nodejs

Cancel Configure

Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the AWS Lambda 'Create function' wizard. The 'Blueprints' tab is active, displaying a list of pre-configured templates. The 'hello-world-python' template is selected. Other visible templates include 'greengrass-hello-world', 'hello-world', 'hello-world-python3', and 'greengrass-hello-world-nodejs'. The interface includes a search bar, export options, and navigation controls.

The screenshot shows the AWS Lambda 'Create function' wizard. The top navigation bar includes 'Services', 'Resource Groups', and user information for Abdul Wahid. The current step is 'Using blueprint hello-world-python'. The 'Basic information' section has the function name set to 'myFunctionName' and the role dropdown set to 'Choose an existing role'. The 'Lambda function code' section shows a pre-configured Python 2.7 function:

```
1 from __future__ import print_function
2
3 import json
4
5 print('Loading function')
6
7
8 def lambda_handler(event, context):
9     #print("Received event: " + json.dumps(event, indent=2))
10    print("Value1 = " + event['key1'])
11    print("Value2 = " + event['key2'])
12    print("Value3 = " + event['key3'])
13    return event['key1'] # Echo back the first key value
14    #raise Exception('Something went wrong')
15
```

At the bottom, a note says '* These fields are required.' and there are 'Cancel', 'Previous', and 'Create function' buttons. The footer includes links for Feedback, English (US), Privacy Policy, and Terms of Use.

The screenshot shows the AWS Lambda console interface for a function named "HelloWorld".

General Information:

- ARN: arn:aws:lambda:us-east-2:042373950390:function:HelloWorld
- Status: Active
- Last modified: 10 minutes ago
- Created: 10 minutes ago

Designer: Shows triggers and resources. Triggers include API Gateway, AWS IoT, CloudWatch Events, CloudWatch Logs, CodeCommit, and CloudSync Triggers. Resources include Amazon CloudWatch Logs and Amazon S3.

Function code: Python 2.7 Handler: lambda_function.lambda_handler

```
File Edit Find View Tools Window
Environment HelloWorld
lambda_function
1 from __future__ import print_function
2
3 import base64
4
5 print('Loading function')
6
7 def lambda_handler(event, context):
8     print("Received event: " + str(event))
9     print("Event key-value pairs: " + str(event['key']))
10    print("Value: " + str(event['value']))
11    print("First key value: " + str(event['key'][0]))
12    print("First key value: " + str(event['key'][0]['first']))
13
14    #raise Exception("Something went wrong")
```

Environment variables: No environment variables defined.

Tags: lambda-console-blueprint, hello-world-python

Execution role: Choose an existing role: service-role/basicSSPermission

Basic settings:

- Description: A starter AWS Lambda function.
- Memory (MB): 128 MB
- Timeout: 3 seconds

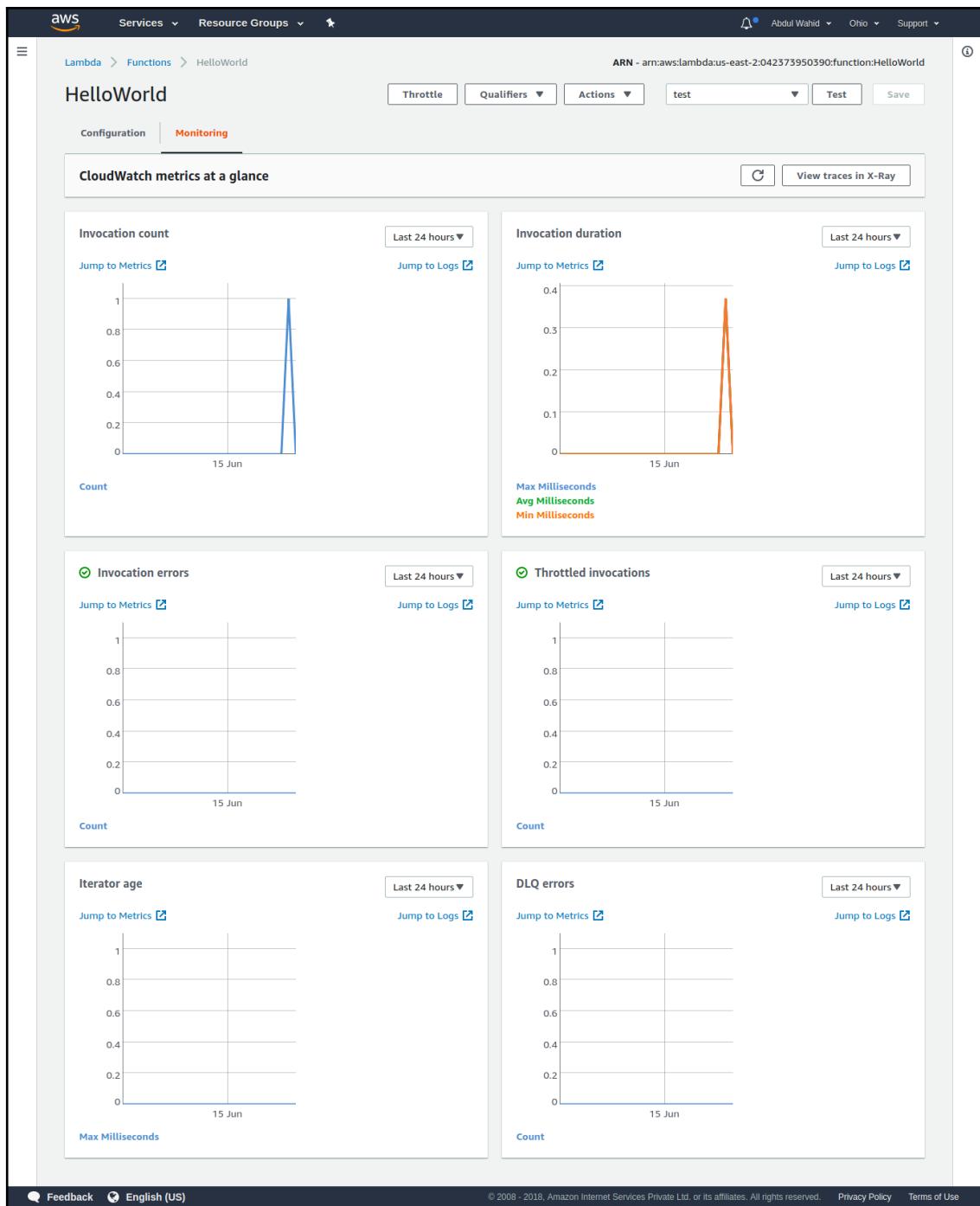
Network: No VPC selected.

Concurrency: Unreserved account concurrency: 1000. Use unreserved account concurrency is selected.

Debugging and error handling:

- DLQ Resource: None
- Enable active tracing: None

Auditing and compliance: This function's invocations can be logged by CloudTrail for operational and risk auditing, governance, and compliance. Visit the CloudTrail console to get started.



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

Hello World



Event name

MyEventName

```
1 + []
2   "key3": "value3",
3   "key2": "value2",
4   "key1": "value1"
5 }
```

HelloWorld

Qualifiers ▾ Actions ▾ helloworld ▾ Test

Execution result: succeeded ([logs](#))

▼ Details

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

```
[ "Hello World"]
```

Summary

Code SHA-256
dvnFe0sFFlg9UWZJHo
dSDU33qveu/DUE760
RUO+WKA=

Request ID
c5d856fa-cd65-11e7-
bf1e-31311818bcd5

Duration
0.40 ms

Billed duration
100 ms

Resources configured
128 MB

Max memory used
18 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: c5d856fa-cd65-11e7-bf1e-31311818bcd5 Version: $LATEST
Received event: {
    "key3": "value3",
    "key2": "value2",
    "key1": "value1"
}
value1 = value1
value2 = value2
value3 = value3
```

HelloWorld Qualifiers Actions helloworld Test

Execution result: failed (logs)

Details

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

```
{ "stackTrace": [ [ "/var/task/lambda_function.py", 10, "lambda_handler", "raise Exception('Exception raised manually.')" ] ], "errorType": "Exception", "errorMessage": "Exception raised manually." }
```

Summary

Code SHA-256
of359p5Ah1hFMOm9
OqDsy6d6ct8lfR6Lvt0
a9W7wp9s=

Request ID
73d23bef-cd68-11e7-
9316-375cf87697ef

Duration
0.79 ms

Billed duration
100 ms

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: 73d23bef-cd68-11e7-9316-375cf87697ef Version: $LATEST
Received event: {
    "key3": "value3",
    "key2": "value2",
    "key1": "value1"
}
Exception raised manually.: Exception
Traceback (most recent call last):
  File "/var/task/lambda_function.py", line 10, in lambda_handler
    raise Exception('Exception raised manually.')
--> raise Exception('Exception raised manually.'))
```

```
lambda_function.py
1 from __future__ import print_function
2
3 import json
4 from datetime import datetime
5
6 print('Loading function')
7
8
9 def lambda_handler(event, context):
10     current_time = datetime.now()
11     print ('Hello World with Request ID {0}'.format(context.aws_request_id))
12     print ("Function executed on {0}"
13           .format(current_time.strftime('%d-%b-%Y at %H:%M:%S')) )
14
```

Screenshot of the AWS Lambda Function Configuration page for the HelloWorld function.

Header: AWS Services Resource Groups ARN - arn:aws:lambda:us-east-2:042373950390:function:HelloWorld

Function Name: HelloWorld

Configuration Tab: Monitoring

Designer:

- Shows the HelloWorld function icon.
- Shows CloudWatch Events trigger configuration required.
- Shows Amazon CloudWatch Logs and Amazon S3 resources the function's role has access to.
- Shows a list of triggers on the left: API Gateway, AWS IoT, CloudWatch Events, CloudWatch Logs, CodeCommit, Cognito Sync Trigger, and DynamoDB.

Configure triggers:

Rule: Pick an existing rule, or create a new one. Create a new rule

Rule name*: hello-world-every-minute

Rule description: Say hello world every minute

Rule type: Trigger your target based on an event pattern, or based on an automated schedule.

Event pattern

Schedule expression

Schedule expression*: rate(1 minute)

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon CloudWatch Events 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).

Buttons: Cancel Add

Footer: Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Time (UTC +00:00)	Message
2017-11-20	
▶ 21:08:04	No older events found at the moment. Retry .
▶ 21:08:04	Loading function
▶ 21:08:04	START RequestId: e6296dfc-ce36-11e7-bac9-3f8e6deaa7e8 Version: \$LATEST
▶ 21:08:04	Hello World with Request ID e6296dfc-ce36-11e7-bac9-3f8e6deaa7e8
▶ 21:08:04	Function executed on 20-Nov-2017 at 21:08:04
▶ 21:08:04	END RequestId: e6296dfc-ce36-11e7-bac9-3f8e6deaa7e8
▶ 21:08:04	REPORT RequestId: e6296dfc-ce36-11e7-bac9-3f8e6deaa7e8 Duration: 0.36 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 18 MB
▶ 21:09:05	START RequestId: 0a616f35-ce37-11e7-9f62-59345c32daa8 Version: \$LATEST
▶ 21:09:05	Hello World with Request ID 0a616f35-ce37-11e7-9f62-59345c32daa8
▶ 21:09:05	Function executed on 20-Nov-2017 at 21:09:05
▶ 21:09:05	END RequestId: 0a616f35-ce37-11e7-9f62-59345c32daa8
▶ 21:09:05	REPORT RequestId: 0a616f35-ce37-11e7-9f62-59345c32daa8 Duration: 0.55 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 18 MB
▶ 21:10:04	START RequestId: 2da8f4b8-ce37-11e7-b64d-6f3aad17b10b Version: \$LATEST
▶ 21:10:04	Hello World with Request ID 2da8f4b8-ce37-11e7-b64d-6f3aad17b10b
▶ 21:10:04	Function executed on 20-Nov-2017 at 21:10:04
▶ 21:10:04	END RequestId: 2da8f4b8-ce37-11e7-b64d-6f3aad17b10b
▶ 21:10:04	REPORT RequestId: 2da8f4b8-ce37-11e7-b64d-6f3aad17b10b Duration: 51.24 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 19 MB
▶ 21:11:05	START RequestId: 51e53be6-ce37-11e7-aa37-2785a702ebce Version: \$LATEST
▶ 21:11:05	Hello World with Request ID 51e53be6-ce37-11e7-aa37-2785a702ebce
▶ 21:11:05	Function executed on 20-Nov-2017 at 21:11:05
▶ 21:11:05	END RequestId: 51e53be6-ce37-11e7-aa37-2785a702ebce
▶ 21:11:05	REPORT RequestId: 51e53be6-ce37-11e7-aa37-2785a702ebce Duration: 39.17 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 19 MB
▶ 21:12:04	START RequestId: 7545c85d-ce37-11e7-97ff-e51d75a5e5a1 Version: \$LATEST
▶ 21:12:04	Hello World with Request ID 7545c85d-ce37-11e7-97ff-e51d75a5e5a1
▶ 21:12:04	Function executed on 20-Nov-2017 at 21:12:04
▶ 21:12:04	END RequestId: 7545c85d-ce37-11e7-97ff-e51d75a5e5a1
▶ 21:12:04	REPORT RequestId: 7545c85d-ce37-11e7-97ff-e51d75a5e5a1 Duration: 28.00 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 19 MB
No newer events found at the moment. Retry .	

```
lambda_function.py
1 from __future__ import print_function
2
3 import json
4
5 print('Loading function')
6
7
8 def lambda_handler(event, context):
9     data = {"message": "Hello World returned in JSON"}
10    headers = {"Content-Type": "application/json"}
11    return {"statusCode": 200, \
12            "headers": headers, \
13            "body": json.dumps(data)}
```

Serviços ▾ Resource Groups ▾

ARN - arn:aws:lambda:us-east-2:042373950390:function:HelloWorld

Throttle Qualifiers Actions test 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

CloudWatch Events Saved API Gateway Configuration required

HelloWorld

Amazon CloudWatch Logs

Amazon S3

Resources the function's role has access to will be shown here

Configure triggers

We'll set up an API Gateway endpoint with a proxy integration type (learn more about the input and output format). Any method (GET, POST, etc.) will trigger your integration. To set up more advanced method mappings or subpath routes, visit the Amazon API Gateway console.

API

Pick an existing API, or create a new one.

Create a new API

API name

Enter a name to uniquely identify your API.

LambdaMicroservice

Deployment stage

The name of your API's deployment stage.

prod

Security

Configure the security mechanism for your API endpoint.

Open

Warning: Your API endpoint will be publicly available and can be invoked by all users.

Lambda will add the necessary permissions for Amazon API Gateway to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

Cancel Add

Feedback English (US)

© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

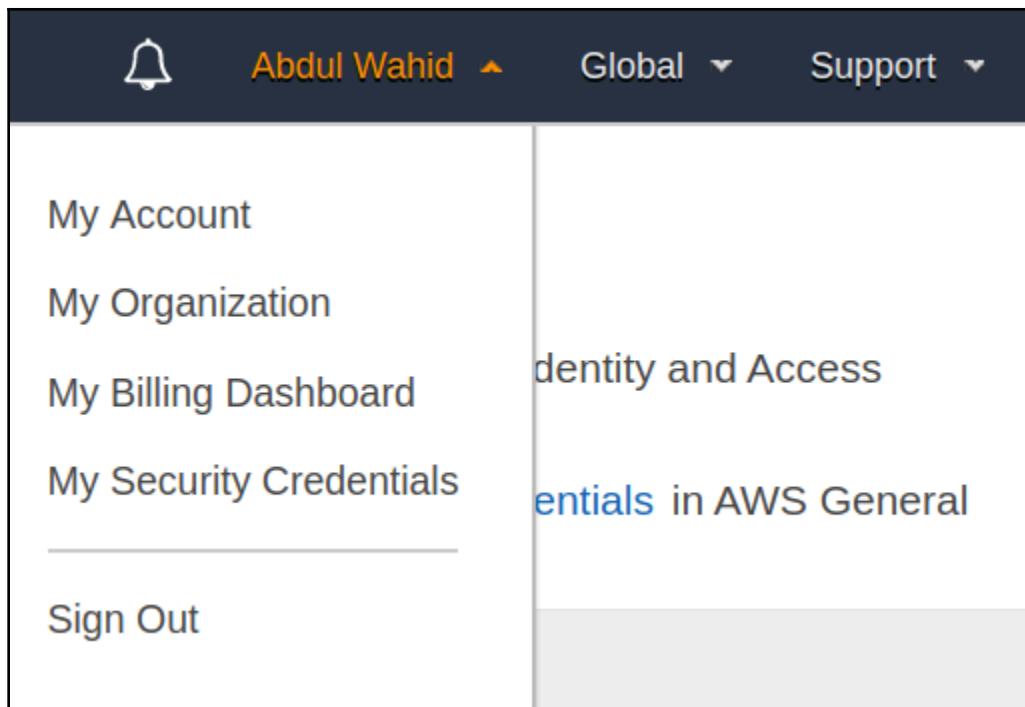
The screenshot shows the AWS Lambda console interface. At the top, the ARN is listed as `arn:aws:lambda:us-east-2:042373950390:function:HelloWorld`. Below the ARN, there are tabs for **Throttle**, **Qualifiers**, **Actions**, **test**, **Test**, and **Save**. The **Configuration** tab is active, and the **Monitoring** tab is visible below it.

In the **Designer** section, under **Add triggers**, the **API Gateway** trigger is selected and marked as **Saved**. Other triggers listed include CloudWatch Events, CloudWatch Logs, CodeCommit, and Cognito Sync Trigger.

The **API Gateway** section displays a single endpoint named **LambdaMicroservice** with the ARN `arn:aws:execute-api:us-east-2:042373950390:cfl6872cxa/*/*/HelloWorld`. The endpoint is **Enabled** and has a **Delete** button.

At the bottom, there are links for **Feedback**, **English (US)**, and **Privacy Policy**.

```
(.env) abdulw@ULTP-711:~/workspace/lambda_poc$ aws configure
AWS Access Key ID [*****QAPA]:
AWS Secret Access Key [*****u98V]:
Default region name [ap-south-1]:
Default output format [json]:
(.env) abdulw@ULTP-711:~/workspace/lambda_poc$ █
```



```
handler.py ✘
1  from __future__ import print_function
2
3  print('Loading function')
4
5
6  def lambda_handler(event, context):
7      return "Hello World ! Response by lambda function."
8
```

```

abdulw@ULTP-711:~/workspace/lambda_poc$ ls
handler.py
abdulw@ULTP-711:~/workspace/lambda_poc$ cat handler.py
from __future__ import print_function

print('Loading function')

def lambda_handler(event, context):
    return "Hello World ! Response by lambda function."

```



```

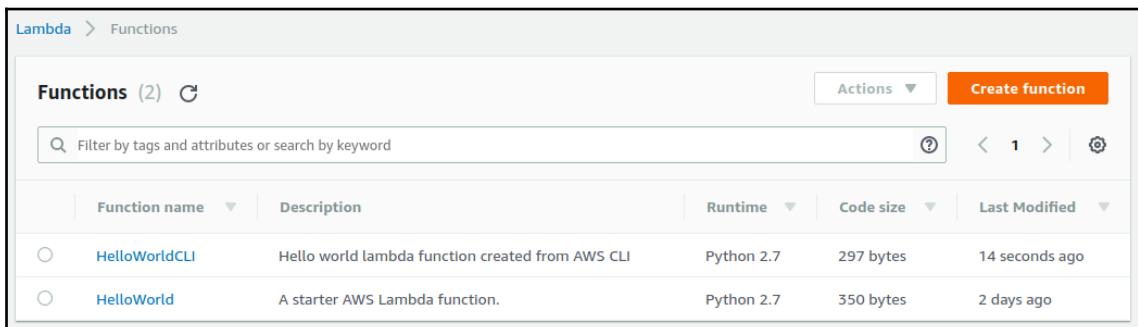
abdulw@ULTP-711:~/workspace/lambda_poc$ zip -r hello_world_package.zip handler.py
  adding: handler.py (deflated 22%)
abdulw@ULTP-711:~/workspace/lambda_poc$ ls
handler.py  hello_world_package.zip
abdulw@ULTP-711:~/workspace/lambda_poc$ █

```

```

abdulw@ULTP-711:~/workspace/lambda_poc$ aws lambda create-function --function-name HelloWorldCLI
--description 'Hello world lambda function created from AWS CLI' --runtime python2.7 --handler
'handler.lambda_handler' --role arn:aws:iam::042373950390:role/service-role/SimpleMicroservicePe
rmission --zip-file fileb://hello_world_package.zip
{
    "FunctionName": "HelloWorldCLI",
    "Role": "arn:aws:iam::042373950390:role/service-role/SimpleMicroservicePermission",
    "LastModified": "2017-11-22T15:53:43.821+0000",
    "Description": "Hello world lambda function created from AWS CLI",
    "Version": "$LATEST",
    "CodeSize": 297,
    "Timeout": 3,
    "FunctionArn": "arn:aws:lambda:us-east-1:042373950390:function:HelloWorldCLI",
    "MemorySize": 128,
    "Runtime": "python2.7",
    "CodeSha256": "UQU7I97ldui98qE9NIGm4KoyBHK7vYXgAUWKvLDEhdk=",
    "Handler": "handler.lambda_handler"
}
abdulw@ULTP-711:~/workspace/lambda_poc$ █

```



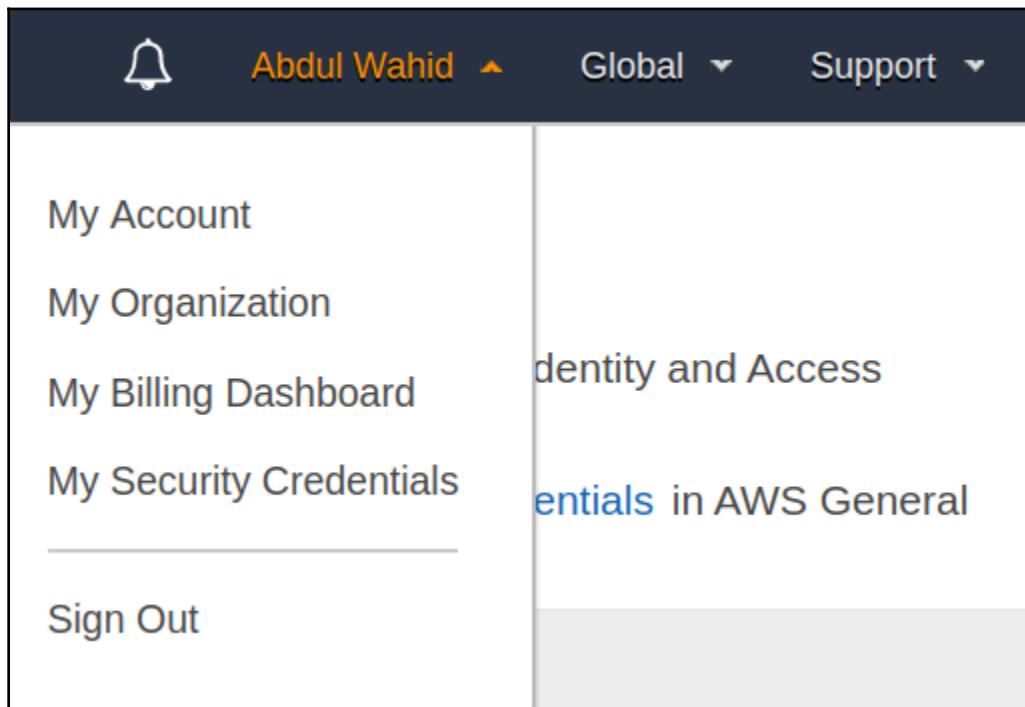
The screenshot shows the AWS Lambda Functions list interface. At the top, there's a navigation bar with 'Lambda > Functions'. Below it is a search bar with placeholder text 'Filter by tags and attributes or search by keyword'. To the right of the search bar are buttons for 'Actions' (with a dropdown arrow) and 'Create function'.

	Function name	Description	Runtime	Code size	Last Modified
<input type="radio"/>	HelloWorldCLI	Hello world lambda function created from AWS CLI	Python 2.7	297 bytes	14 seconds ago
<input type="radio"/>	HelloWorld	A starter AWS Lambda function.	Python 2.7	350 bytes	2 days ago

```
abdulw@ULTP-711:~/workspace/lambda_poc$ ls
handler.py  hello_world_package.zip
abdulw@ULTP-711:~/workspace/lambda_poc$ aws lambda invoke --function-name HelloWorldCLI lambda_output.txt
{
    "StatusCode": 200
}
abdulw@ULTP-711:~/workspace/lambda_poc$ ls
handler.py  hello_world_package.zip  lambda_output.txt
abdulw@ULTP-711:~/workspace/lambda_poc$ cat lambda_output.txt && echo
"Hello World ! Response by lambda function."
abdulw@ULTP-711:~/workspace/lambda_poc$ █
```

Chapter 2: Getting Started with Zappa

```
(.env) abdulw@ULTP-711:~/workspace/lambda_poc$ aws configure
AWS Access Key ID [*****QAPA]:
AWS Secret Access Key [*****u98V]:
Default region name [ap-south-1]:
Default output format [json]
(.env) abdulw@ULTP-711:~/workspace/lambda_poc$ █
```



```
hello.py x

1  from bottle import Bottle, run
2
3  app = Bottle()
4
5  @app.route('/hello')
6  def hello():
7      return "Hello World!\n"
8
```

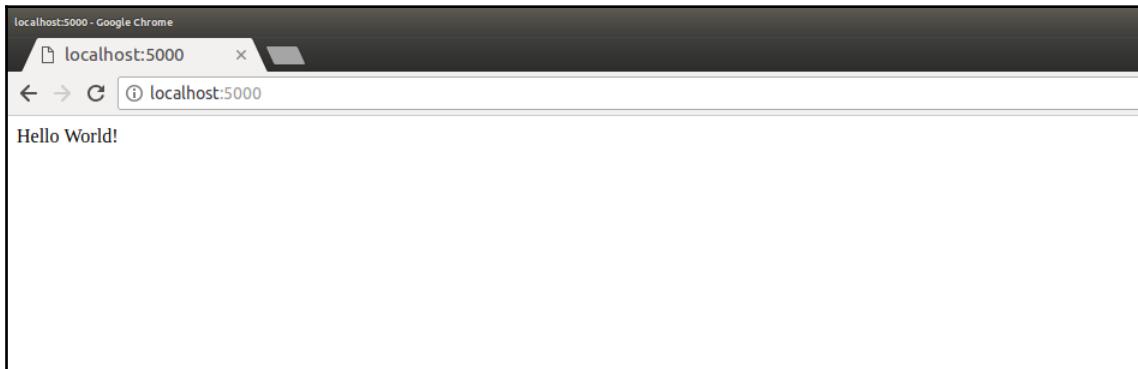
```
(env) abdulw@ULTP-711:~/workspace/lambda_bottle_poc$ ls
env hello.py
(env) abdulw@ULTP-711:~/workspace/lambda_bottle_poc$ zappa init

  
Welcome to Zappa!  
Zappa is a system for running server-less Python web applications on AWS Lambda and AWS API Gateway.  
This 'init' command will help you create and configure your new Zappa deployment.  
Let's get started!  
Your Zappa configuration can support multiple production stages, like 'dev', 'staging', and 'production'.  
What do you want to call this environment (default 'dev'):  
AWS Lambda and API Gateway are only available in certain regions. Let's check to make sure you have a profile set up in one that will work.  
Okay, using profile default!  
Your Zappa deployments will need to be uploaded to a private s3 bucket.  
If you don't have a bucket yet, we'll create one for you too.  
What do you want call your bucket? (default 'zappa-2o2zd8dg4'):br/>What's the modular path to your app's function?  
This will likely be something like 'your_module.app'.  
Where is your app's function?: hello.app  
You can optionally deploy to all available regions in order to provide fast global service.  
If you are using Zappa for the first time, you probably don't want to do this!  
Would you like to deploy this application globally? (default 'n') [y/n/(p)primary]:  
Okay, here's your zappa_settings.json:  
[{"dev": {"app_function": "hello.app", "aws_region": "ap-south-1", "profile_name": "default", "project_name": "lambda-bottle-p", "runtime": "python3.6", "s3_bucket": "zappa-2o2zd8dg4"}}  
Does this look okay? (default 'y') [y/n]: y  
Done! Now you can deploy your Zappa application by executing:  
$ zappa deploy dev  
After that, you can update your application code with:  
$ zappa update dev  
To learn more, check out our project page on GitHub here: https://github.com/Miserlou/Zappa  
and stop by our Slack channel here: https://slack.zappa.io  
Enjoy!,  
~ Team Zappa!  
(env) abdulw@ULTP-711:~/workspace/lambda_bottle_poc$
```

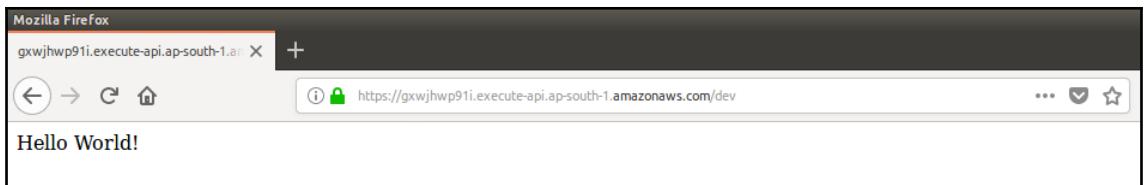
```
[env] abdulw@ULTP-711:~/workspace/lambda_bottle_poc$ zappa deploy dev
Calling deploy for stage dev..
Downloading and installing dependencies..
    sqlite=python36: Using precompiled lambda package
Packaging project as zip.
Uploading lambda-bottle-p-dev-1512940199.zip (5.2MiB)..          | 5.41M/5.41M [00:02<00:00, 2.54MB/s]
100%|██████████| 5.41M/5.41M [00:02<00:00, 2.54MB/s]
Scheduling..
Scheduled lambda-bottle-p-dev-zappa-keep-warm-handler.keep_warm_callback with expression rate(4 minutes)!
Uploading lambda-bottle-p-dev-template-1512940205.json (1.6KiB)..          | 1.65K/1.65K [00:00<00:00, 8.29KB/s]
100%|██████████| 1.65K/1.65K [00:00<00:00, 8.29KB/s]
Waiting for stack lambda-bottle-p-dev to create (this can take a bit)..          | 4/4 [00:12<00:00, 4.32s/res]
100%|██████████| 4/4 [00:12<00:00, 4.32s/res]
Deploying API Gateway..
Deployment complete!: https://071h4br4e0.execute-api.ap-south-1.amazonaws.com/dev
(env) abdulw@ULTP-711:~/workspace/lambda_bottle_poc$
```

Chapter 3: Building a Flask Application with Zappa

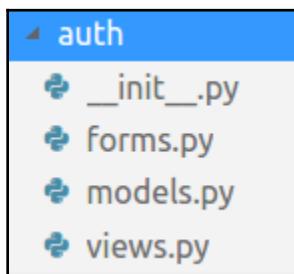
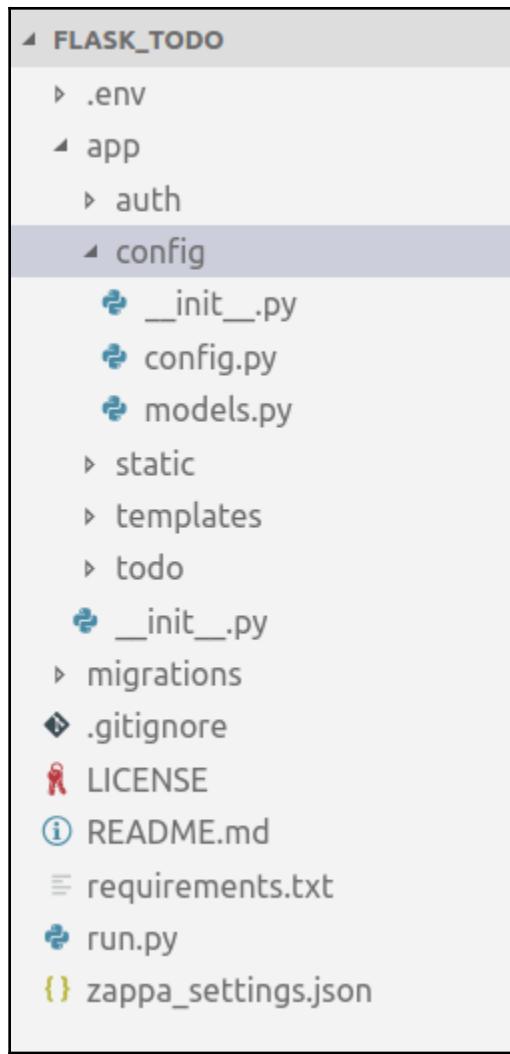
```
abdulw@ULTP-711:~$ mkdir flask_todo
abdulw@ULTP-711:~$ cd flask_todo/
abdulw@ULTP-711:~/flask_todo$ virtualenv .env -p python3.6
Running virtualenv with interpreter /usr/bin/python3.6
Using base prefix '/usr'
New python executable in /home/abdulw/flask_todo/.env/bin/python3.6
Also creating executable in /home/abdulw/flask_todo/.env/bin/python
Installing setuptools, pip, wheel...done.
abdulw@ULTP-711:~/flask_todo$
```



```
(.env) abdulw@ULTP-711:~/flask_todo$ zappa deploy dev
Calling deploy for stage dev..
Creating flask-todo-dev-ZappaLambdaExecutionRole IAM Role..
Creating zappa-permissions policy on flask-todo-dev-ZappaLambdaExecutionRole IAM Role.
Downloading and installing dependencies..
    - soliter/python36: Using precompiled lambda package
Packaging project as zip..
Uploading flask-todo-dev-1514664256.zip (5.6MB)..[00:02<00:00, 2.52MB/s]
Scheduling..
Scheduled flask-todo-dev-zappa-keep-warm-handler.keep_warm_callback with expression rate(4 minutes)..
Uploading flask-todo-dev-template-1514664316.json (1.6KiB)..[00:00<00:00, 4.40KB/s]
Waiting for stack flask-todo-dev to create (this can take a bit)..[00:12<00:00, 4.29s/res]
Deploying API Gateway..
Deployment complete!: https://gxwjhwpp91i.execute-api.ap-south-1.amazonaws.com/dev
(.env) abdulw@ULTP-711:~/flask_todo$
```

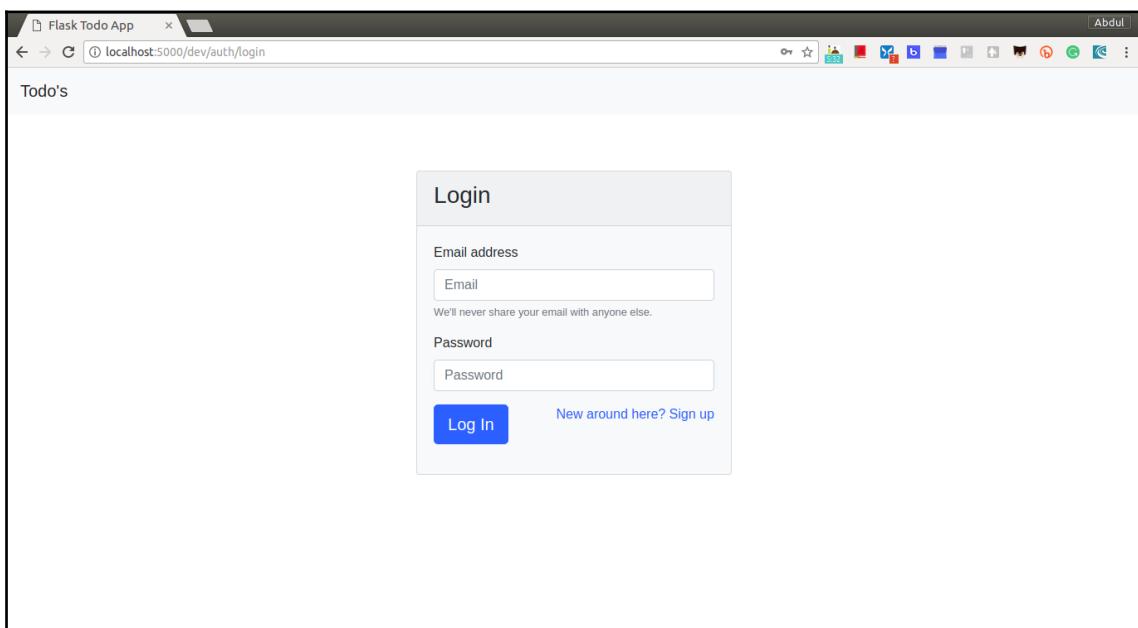
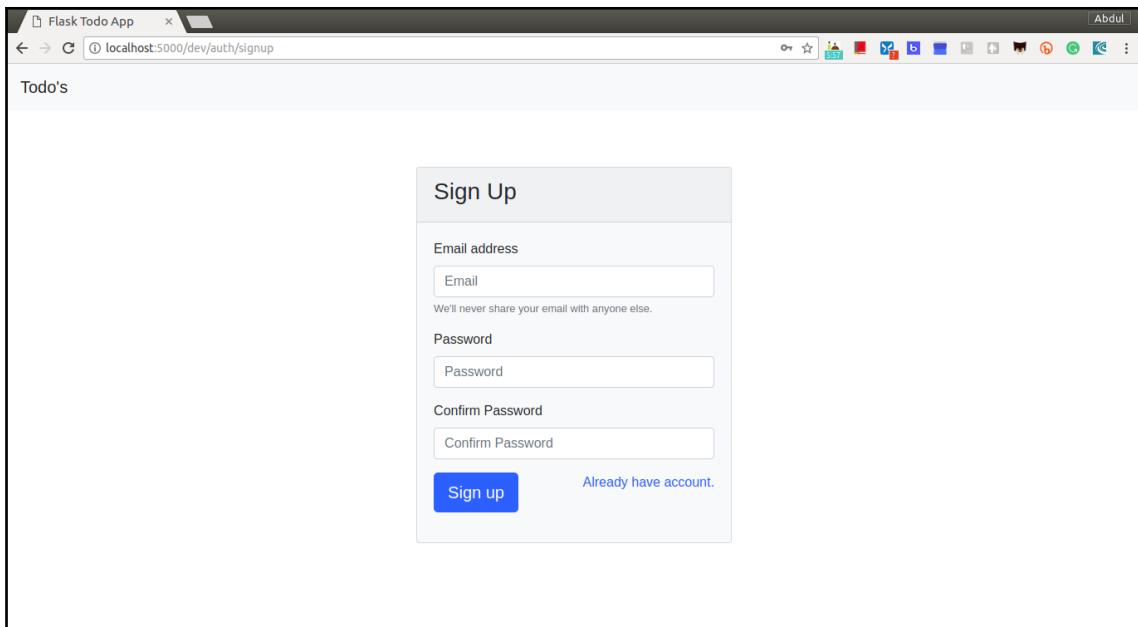


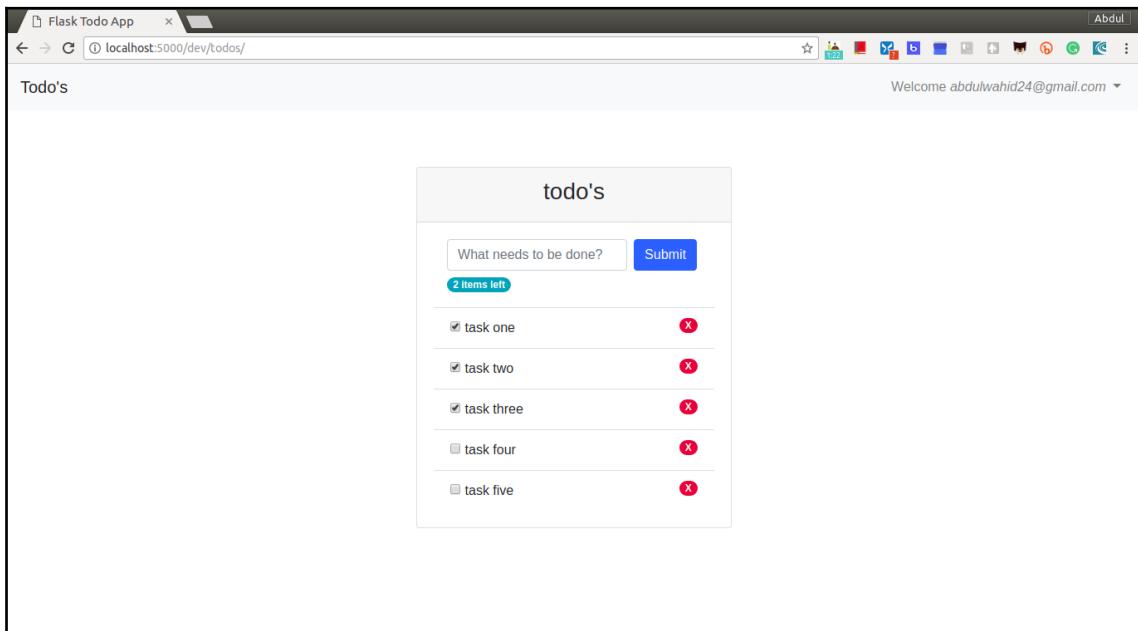
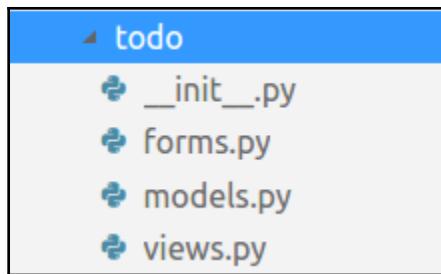
```
abdulw@ULTP-711:~/workspace/flask_todo(master)$ virtualenv .env -p python3.6
Running virtualenv with interpreter /usr/bin/python3.6
Using base prefix '/usr'
New python executable in /home/abdulw/workspace/flask_todo/.env/bin/python3.6
Also creating executable in /home/abdulw/workspace/flask_todo/.env/bin/python
Installing setuptools, pip, wheel...done.
abdulw@ULTP-711:~/workspace/flask_todo(master)$ source .env/bin/activate
(.env) abdulw@ULTP-711:~/workspace/flask_todo(master)$
```



```
▲ templates
  ▾ auth
    ◊ login.html
    ◊ signup.html
  ▾ todo
    ◊ list.html
    ◊ base.html
    ◊ navbar.html
```

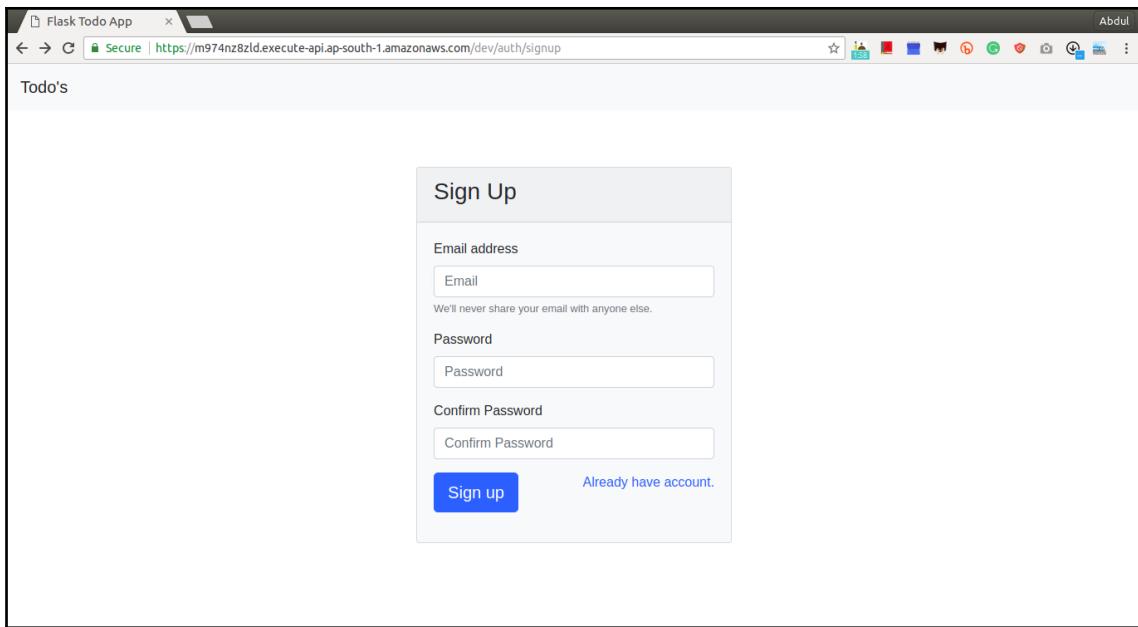
```
▲ static
  ▾ bootstrap
    ▾ css
      # bootstrap.min.css
      # bootstrap.min.css.map
    ▾ js
      JS bootstrap.min.js
      JS bootstrap.min.js.map
  ▾ jquery
    JS jquery-3.3.1.min.js
  ▾ popper
    JS popper.min.js
  ▾ todo
    JS list.js
```





```
(.env) abdulw@ULTR-711:~/workspace/flask_todo(master)$ zappa init

Welcome to Zappa!
Zappa is a system for running server-less Python web applications on AWS Lambda and AWS API Gateway.
This 'init' command will help you create and configure your new Zappa deployment.
Let's get started!
Your Zappa configuration can support multiple production stages, like 'dev', 'staging', and 'production'.
What do you want to call this environment (default 'dev'):
AWS Lambda and API Gateway are only available in certain regions. Let's check to make sure you have a profile set up in one that will work.
Okay, using profile default!
Your Zappa deployments will need to be uploaded to a private S3 bucket.
If you don't have a bucket yet, we'll create one for you too.
What do you want call your bucket? (default 'zappa-zfluekcq'):
It looks like this is a Flask application.
What's the modular path to your app's function?
This will likely be something like 'your_module.app'.
We discovered: app.__init__.app
Where is your app's function? (default 'app.__init__.app'): run.app
You can optionally deploy to all available regions in order to provide fast global service.
If you are using Zappa for the first time, you probably don't want to do this!
Would you like to deploy this application globally? (default 'n') [y/n/(p)imary]: n
Okay, here's your zappa_settings.json:
{
    "dev": {
        "app_function": "run.app",
        "aws_region": "ap-south-1",
        "profile_name": "default",
        "project_name": "flask-todo",
        "runtime": "python3.6",
        "s3_bucket": "zappa-zfluekcq"
    }
}
Does this look okay? (default 'y') [y/n]: y
Done! Now you can deploy your Zappa application by executing:
$ zappa deploy dev
After that, you can update your application code with:
$ zappa update dev
To learn more, check out our project page on GitHub here: https://github.com/miserlou/zappa
and stop by our Slack channel here: https://slack.zappa.io
Enjoy!,
~ Team Zappa!
(.env) abdulw@ULTR-711:~/workspace/flask_todo(master)$
```



Chapter 4: Building a Flask-Based REST API with Zappa



Request

Method: POST Request URL: https://jrzw1zpd.execute-api.ap-south-1.amazonaws.com/dev/auth/signup

SEND ⋮

Parameters ^

Headers Body Variables

Body content type application/json Editor view JSON visual editor

email abdul.wahid24@gmail.com ↎ ⏎
password abdul123 ↎ ⏎

+ ADD PROPERTY

200 OK 342.70 ms DETAILS ▾

Selected environment: Default ⓘ

The screenshot shows a POST request to the '/auth/signup' endpoint. The response status is 200 OK with a duration of 342.70 ms. The response body is a JSON object with a single key 'message' containing the value 'Sign up successfully'.

Request

Method: POST Request URL: https://jrzw1zpd.execute-api.ap-south-1.amazonaws.com/dev/auth/login

SEND ⋮

Parameters ^

Headers Body Variables

Body content type application/json Editor view JSON visual editor

email abdul.wahid24@gmail.com ↎ ⏎
password abdul123 ↎ ⏎

+ ADD PROPERTY

200 OK 482.80 ms DETAILS ▾

Selected environment: Default ⓘ

The screenshot shows a POST request to the '/auth/login' endpoint. The response status is 200 OK with a duration of 482.80 ms. The response body is a JSON object with a single key 'token' containing a long string of characters.

Request

Method: GET Request URL: https://jrzw1zpd1.execute-api.ap-south-1.amazonaws.com/dev/todos/

SEND ⋮

Parameters ^

Headers Variables

Header name: Content-Type Header value: application/json

ADD HEADER

A Headers size: 30 bytes

401 Unauthorized 190.90 ms DETAILS

Header name: Content-Type Header value: application/json

{ "description": "Request does not contain an access token", "error": "Authorization Required", "status_code": 401 }

Selected environment: Default

This screenshot shows a POST request to the '/todos' endpoint of an AWS Lambda function. The 'Content-Type' header is set to 'application/json'. The response is a 401 Unauthorized status with a message indicating that the request does not contain an access token.

Request

Method: GET Request URL: https://jrzw1zpd1.execute-api.ap-south-1.amazonaws.com/dev/todos/

SEND ⋮

Parameters ^

Headers Variables

Header name: Content-Type Header value: application/json

Header name: Authorization Header value: JWT eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE1MjkzODE0NDQsImhdCl6MTUyOTM4MTE0NCw

ADD HEADER

A Headers size: 218 bytes

200 OK 164.50 ms DETAILS

Header name: Content-Type Header value: application/json

[Array[0]]

Selected environment: Default

This screenshot shows the same POST request to the '/todos' endpoint, but this time it includes an 'Authorization' header with a valid JWT token. The response is a 200 OK status with an empty array as the body.

Request

Method POST Request URL https://jrzw1zpd1.execute-api.ap-south-1.amazonaws.com/dev/todos/

Parameters ^

Headers Body Variables

Body content type application/json Editor view JSON visual editor

title task one

+ ADD PROPERTY

201 Created 256.50 ms DETAILS

Selected environment: Default

```
{ "id": 3, "title": "task one", "created_by": "abdul.wahid24@gmail.com", "status": "open" }
```

Request

Method POST Request URL https://jrzw1zpd1.execute-api.ap-south-1.amazonaws.com/dev/todos/

Parameters ^

Headers Body Variables

Body content type application/json Editor view JSON visual editor

+ ADD PROPERTY

400 Bad Request 154.10 ms DETAILS

Selected environment: Default

```
{ "message": { "title": "Missing required parameter in the JSON body or the post body or the query string" } }
```

Request

Method: GET Request URL: https://jrzw1zpd.execute-api.ap-south-1.amazonaws.com/dev/todos/5/

SEND ⋮

Parameters ^

Headers Variables

Header name: Content-Type Header value: application/json

Header name: Authorization Header value: JWT eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOjE1MjkzODE4MTEsImhdCl6MTUyOTM4MTUxMSw

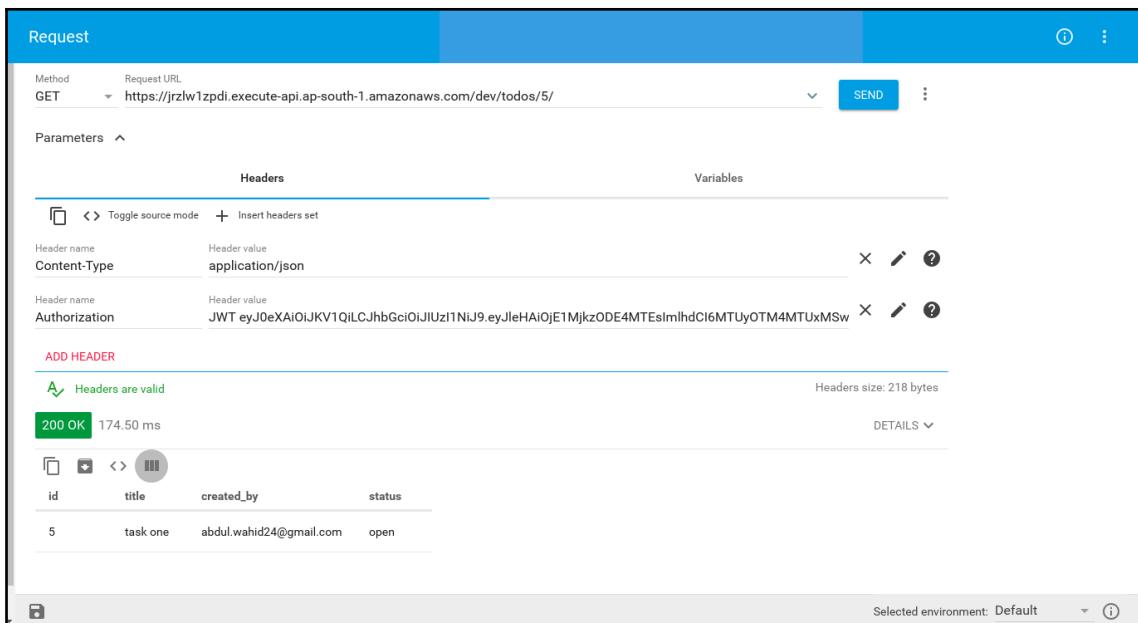
ADD HEADER

Headers are valid Headers size: 218 bytes

200 OK 174.50 ms DETAILS ▾

id	title	created_by	status
5	task one	abdul.wahid24@gmail.com	open

Selected environment: Default ⓘ



Request

Method: PATCH Request URL: https://jrzw1zpd.execute-api.ap-south-1.amazonaws.com/dev/todos/5/

SEND ⋮

Parameters ^

Headers Body Variables

Body content type: application/json Editor view: JSON visual editor

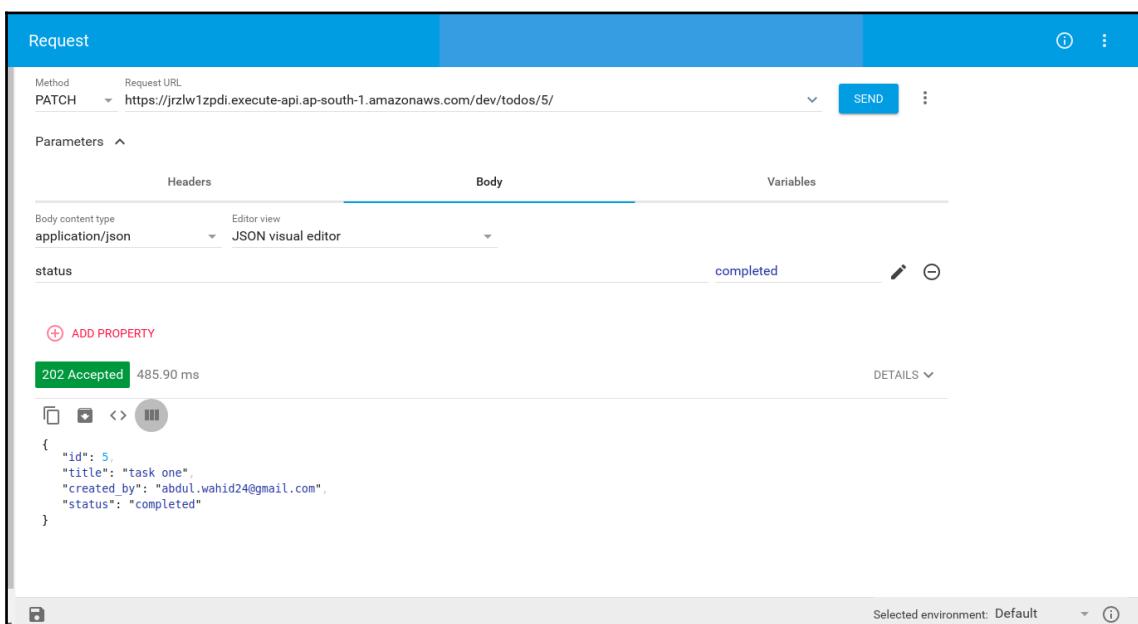
status: completed

+ ADD PROPERTY

202 Accepted 485.90 ms DETAILS ▾

{ "id": 5, "title": "task one", "created_by": "abdul.wahid24@gmail.com", "status": "completed" }

Selected environment: Default ⓘ



Request

Method: PATCH Request URL: https://jrzw1zpd1.execute-api.ap-south-1.amazonaws.com/dev/todos/5/

SEND ⋮

Parameters ^

Headers Body Variables

Body content type application/json Editor view JSON visual editor

+ ADD PROPERTY

400 Bad Request 167.60 ms DETAILS

Selected environment: Default ⓘ

```
{ "message": { "status": "Bad choice: Missing required parameter in the JSON body or the post body or the query string. Valid choices are 'open' or 'completed'." }}
```

Request

Method: DELETE Request URL: https://jrzw1zpd1.execute-api.ap-south-1.amazonaws.com/dev/todos/5/

SEND ⋮

Parameters ^

Headers Body Variables

Body content type application/json Editor view JSON visual editor

+ ADD PROPERTY

204 No Content 273.90 ms DETAILS

Selected environment: Default ⓘ

Chapter 5: Building a Django Application with Zappa

```
abdulw@ULTP-711:~/workspace/book/chapter_5(master)$ pipenv
Usage: pipenv [OPTIONS] COMMAND [ARGS]...

Options:
  --update      Update Pipenv & pip to latest.
  --where       Output project home information.
  --venv        Output virtualenv information.
  --py          Output Python interpreter information.
  --envs        Output Environment Variable options.
  --rm          Remove the virtualenv.
  --bare        Minimal output.
  --completion Output completion (to be eval'd).
  --man         Display manpage.
  --three / --two Use Python 3/2 when creating virtualenv.
  --python TEXT Specify which version of Python virtualenv should use.
  --site-packages Enable site-packages for the virtualenv.
  --jumbotron An easter egg, effectively.
  --version     Show the version and exit.
  -h, --help    Show this message and exit.

Usage Examples:
  Create a new project using Python 3.6, specifically:
  $ pipenv --python 3.6

  Install all dependencies for a project (including dev):
  $ pipenv install --dev

  Create a lockfile containing pre-releases:
  $ pipenv lock --pre

  Show a graph of your installed dependencies:
  $ pipenv graph

  Check your installed dependencies for security vulnerabilities:
  $ pipenv check

  Install a local setup.py into your virtual environment/Pipfile:
  $ pipenv install -e .

Commands:
  check      Checks for security vulnerabilities and against PEP 508 markers
             provided in Pipfile.
  graph      Displays currently-installed dependency graph information.
  install    Installs provided packages and adds them to Pipfile, or (if none
             is given), installs all packages.
  lock       Generates Pipfile.lock.
  open       View a given module in your editor.
  run        Spawns a command installed into the virtualenv.
  shell      Spawns a shell within the virtualenv.
  uninstall  Un-installs a provided package and removes it from Pipfile.
  update    Uninstalls all packages, and re-installs package(s) in [packages]
            to latest compatible versions.
```

```

abdulw@ULTP-711:~/workspace/book/chapter_5(master)$ pipenv --python python3.6
Virtualenv already exists!
Removing existing virtualenv...
Creating a virtualenv for this project...
Using /usr/bin/python3.6 to create virtualenv...
Running virtualenv with interpreter /usr/bin/python3.6
Using base prefix '/usr'
New python executable in /home/abdulw/.local/share/virtualenvs/chapter_5-N1x3-kJk/bin/python3.6
Also creating executable in /home/abdulw/.local/share/virtualenvs/chapter_5-N1x3-kJk/bin/python
Installing setuptools, pip, wheel...done.

Virtualenv location: /home/abdulw/.local/share/virtualenvs/chapter_5-N1x3-kJk
Creating a Pipfile for this project...
abdulw@ULTP-711:~/workspace/book/chapter_5(master)$ pipenv install django
Installing django...
Collecting django
  Downloading Django-2.0.3-py3-none-any.whl (7.1MB)
Collecting pytz (from django)
  Using cached pytz-2018.3-py2.py3-none-any.whl
Installing collected packages: pytz, django
Successfully installed django-2.0.3 pytz-2018.3

Adding django to Pipfile's [packages]...
Locking [dev-packages] dependencies...
Locking [packages] dependencies...
Updated Pipfile.lock (374a8f)!

(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5(master)$ django-admin.py startproject imageGalleryProject
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5(master)$ ls
imageGalleryProject Pipfile Pipfile.lock
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5(master)$ cd imageGalleryProject/
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ ls
imageGalleryProject manage.py
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ python manage.py startapp gallery
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ ls
gallery imageGalleryProject manage.py
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ tree
.
├── gallery
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   └── migrations
│       └── __init__.py
│   ├── models.py
│   ├── tests.py
│   └── views.py
└── imageGalleryProject
    ├── __init__.py
    ├── __pycache__
    │   └── __init__.cpython-36.pyc
    ├── settings.py
    ├── urls.py
    └── wsgi.py

2 directories, 14 files
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$

```

```
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ python manage.py makemigrations
Migrations for 'gallery':
  - Create model PhotoGallery
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, gallery, sessions
Running migrations:
  Applying gallery.0001_initial... OK
(chapter_5-N1x3-kJk) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$
```

Django administration

WELCOME, ABDULWAHID [VIEW SITE / CHANGE PASSWORD / LOG OUT](#)

Home > Gallery > Photo albums > Add photo album

Add photo album

Album name:

PHOTOS

IMAGE	IMAGE THUMBNAIL	DELETE?
<input type="button" value="Choose file"/> No file chosen		DELETE?
+ Add another Photo		

[Save and add another](#) [Save and continue editing](#) [SAVE](#)

Django administration

WELCOME, ABDULWAHID [VIEW SITE / CHANGE PASSWORD / LOG OUT](#)

Home > Gallery > Photo albums

✓ The photo album "Sample Screenshots" was added successfully.

Select photo album to change [ADD PHOTO ALBUM +](#)

Action: 0 of 1 selected

<input type="checkbox"/> PHOTO ALBUM
<input type="checkbox"/> Sample Screenshots

1 photo album

Django administration

WELCOME, ABDULWAHID VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Gallery > Photo albums > Sample Screenshots

Change photo album

Album name: Sample Screenshots

PHOTOS

IMAGE

Screenshot_from_2018-03-07_14-35-19.png

Currently: Sample Screenshots/Screenshot_from_2018-03-07_14-35-19.png

Change: Choose file No file chosen

Screenshot_from_2018-02-21_15-58-41.png

Currently: Sample Screenshots/Screenshot_from_2018-02-21_15-58-41.png

Change: Choose file No file chosen

Choose file No file chosen

+ Add another Photo

Delete

Save and add another Save and continue editing SAVE

AWS CloudFront M X Abdul

Secure | https://console.aws.amazon.com/cloudfront/home?region=ap-south-1#

Services Resource Groups

Distributions

Create Distribution Distribution Settings Delete Enable Disable

Viewing : Any Delivery Method Any State

CloudFront Distributions

Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last Modified
Web	E3R0PUFQH5PHGH	dys62g3nirogi.clou	-	qa.conjuror.in,	-	Deployed	Enabled	2018-02-14 03:02

Viewing 1 to 1 of 1 items

Feedback English (US)

© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the AWS CloudFront console with the URL <https://console.aws.amazon.com/cloudfront/home?region=ap-south-1#create-distribution>. The user is at Step 1: Select delivery method. The 'Web' delivery method is selected, indicated by a blue border around its title. Below it, there's a list of benefits for using a web distribution. A note says you can add more origins later. There are 'Get Started' and 'Cancel' buttons. The top navigation bar includes services like Lambda, S3, and CloudWatch.

Step 1: Select delivery method

Step 2: Create distribution

Select a delivery method for your content.

Web

Create a web distribution if you want to:

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

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

RTMP

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

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

Get Started **Cancel**

Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Create Distribution

Step 1: Select delivery method

Step 2: Create distribution

Origin Settings

Origin Domain Name:

Origin Path:

Origin ID:

Origin Custom Headers: Header Name: Value:

Default Cache Behavior Settings

Path Pattern: Default (*)

Viewer Protocol Policy: * HTTP and HTTPS
Redirect HTTP to HTTPS
HTTP Only

Allowed HTTP Methods: * GET, HEAD
GET, HEAD, OPTIONS
GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Field-level Encryption Config:

Cached HTTP Methods: GET, HEAD (Cached by default)

Cache Based on Selected Request Headers: None (Improves Caching)

Object Caching: * Use Origin Cache Headers
Customize

Minimum TTL: 0

Maximum TTL: 31536000

Default TTL: 86400

Forward Cookies: None (Improves Caching)

Query String Forwarding and Caching: None (Improves Caching)

Smooth Streaming: * Yes
* No

Restrict Viewer Access (Use Signed URLs or Signed Cookies): * Yes
* No

Compress Objects Automatically: * Yes
* No

Lambda Function Associations: Event Type:
Lambda Function ARN:

Distribution Settings

Price Class: Use All Edge Locations (Best Performance)

AWS WAF Web ACL: None

Alternate Domain Names (CNAMEs):

SSL Certificate: * Default CloudFront Certificate (*.cloudfront.net)
Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as https://1111111111111111.cloudfront.net/logo.jpg). Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.

Custom SSL Certificate (example.com):
Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.example.com/logo.jpg. You can use a certificate from AWS Certificate Manager (ACM) in the US East (N. Virginia) Region, or you can use a certificate stored in IAM.

Learn more about using custom SSL/TLS certificate with CloudFront.
Learn more about using ACM.

Supported HTTP Versions: * HTTP2, HTTP/1.1, HTTP/1.0
HTTP/1.1, HTTP/1.0

Default Root Object:

Logging: * On
* Off

Bucket for Logs:

Log Prefix:

Cookie Logging: * On
* Off

Enable IPv6: Learn more

Comment:

Distribution State: * Enabled
* Disabled

Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

```
(chapter_5-Nix3-k3k) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ zappa deploy dev
Calling deploy for stage dev..
Creating zappaLambdaExecutionRole IAM Role..
Creating zappaLambdaExecutionRole IAM Role..
Creating zappa-permissions policy on ImageGalleryPro-dev-ZappaLambdaExecutionRole IAM Role.
Downloading and installing dependencies...
- pillow==5.0.0: Using locally cached manylinux wheel
- lazy-object-proxy==1.3.1: Using locally cached manylinux wheel
- sqlite==python36: Using precompiled lambda package
Packaging project as zip..
Uploading ImageGalleryPro-dev-1521320794.zip (22.5MB)...
100% [██████████] 23.6M/23.6M
M [00:44<00:00, 461KB/s]
Scheduling...
Scheduled ImageGalleryPro-dev-zappa-keep-warm-handler.keep_warm_callback with expression rate(4 minutes)!
Uploading ImageGalleryPro-dev-template-1521320853.json (1.6KB)...
100% [██████████] 1.65K/1.65K
[00:00<00:00, 17.5KB/s]
Waiting for stack ImageGalleryPro-dev to create (this can take a bit)...
100% [██████████] 4/4 [00:15<00:00, 4.48B/res]
Deploying API Gateway...
Deployment complete! https://cfsla2gds0.execute-api.ap-south-1.amazonaws.com/dev
(chapter_5-Nix3-k3k) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$
```

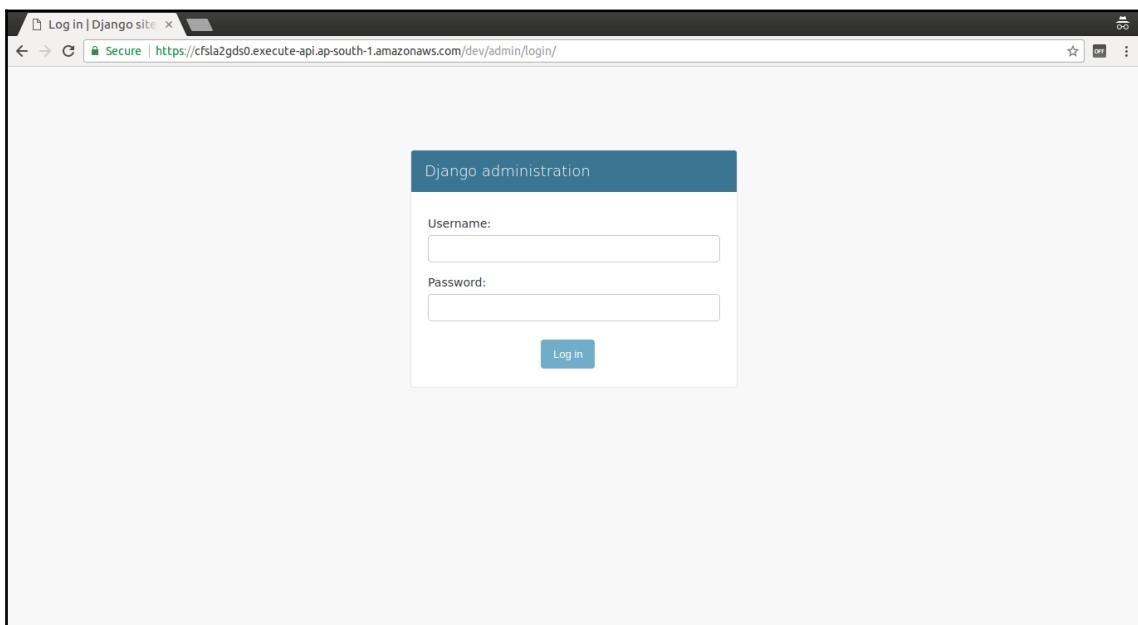
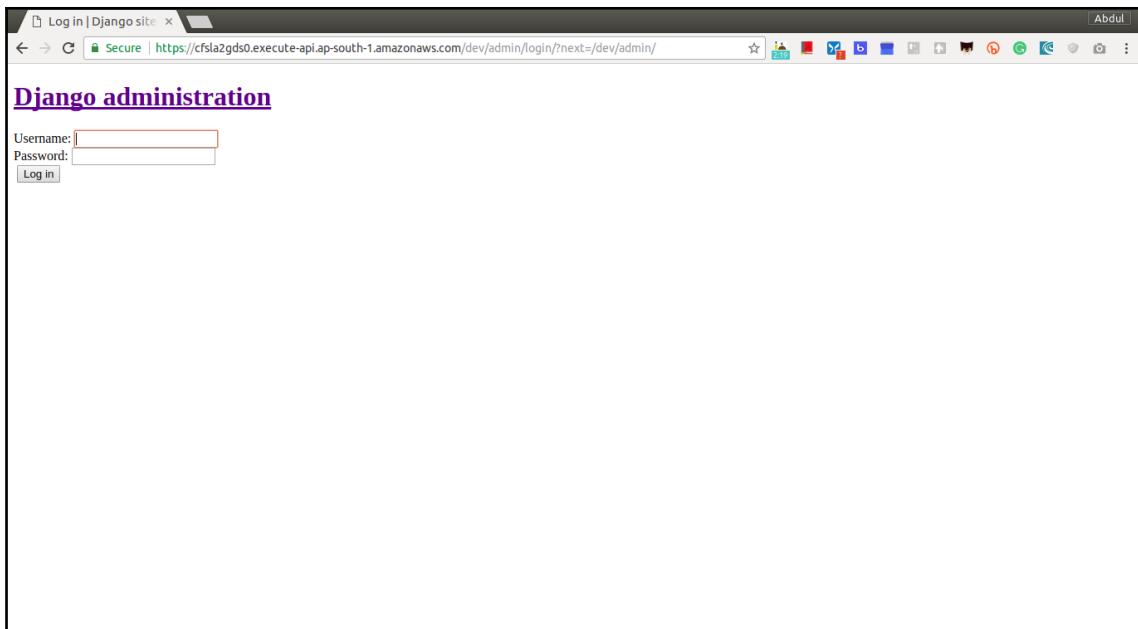
DisallowedHost at /

Invalid HTTP_HOST header: 'cfsla2gds0.execute-api.ap-south-1.amazonaws.com'. You may need to add 'cfsla2gds0.execute-api.ap-south-1.amazonaws.com' to ALLOWED_HOSTS.

Request Method: GET
 Request URL: https://cfsla2gds0.execute-api.ap-south-1.amazonaws.com/dev
 Django Version: 2.0.3
 Exception Type: DisallowedHost
 Exception Value: Invalid HTTP_HOST header: 'cfsla2gds0.execute-api.ap-south-1.amazonaws.com'. You may need to add 'cfsla2gds0.execute-api.ap-south-1.amazonaws.com' to ALLOWED_HOSTS.
 Exception Location: /var/task/django/http/request.py in get_host, line 105
 Python Executable: /var/lang/bin/python3.6
 Python Version: 3.6.1
 Python Path: ['/var/task', '/var/runtime/awslambda', '/var/runtime', '/var/lang/lib/python36.zip', '/var/lang/lib/python3.6', '/var/lang/lib/python3.6/lib-dynload', '/var/lang/lib/python3.6/site-packages', '/var/lang/lib/python3.6/site-packages']
 Server time: Sat, 17 Mar 2018 21:08:46 +0000

Traceback [Switch to copy-and-paste view](#)

```
/var/task/django/core/handlers/exception.py in inner
    35.         response = get_response(request)
    ...
  ▶ Local vars
/var/task/django/utils/deprecation.py in __call__
    93.         response = self.process_request(request)
    ...
  ▶ Local vars
/var/task/django/middleware/common.py in process_request
    55.         host = request.get_host()
    ...
  ▶ Local vars
```



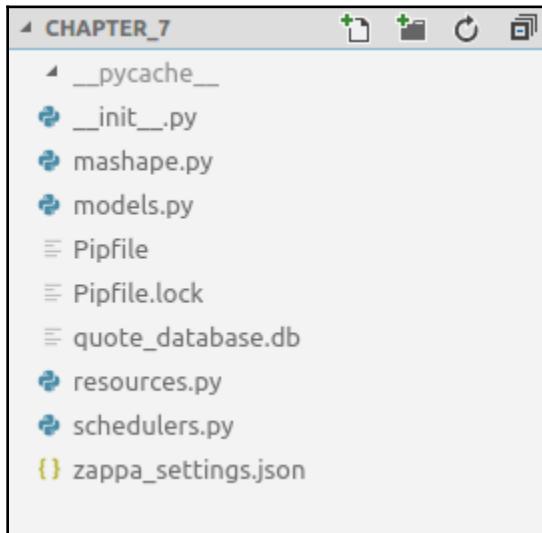
```
(imageGalleryProject-KAW6aJSD) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ zappa manage dev showmigrations admin
[START] RequestId: bcc908da-3bde-11e8-8f67-bb3b239cecd Version: $LATEST
[DEBUG] 2018-04-09T10:14:07.141Z bcc908da-3bde-11e8-8f67-bb3b239cecd Zappa Event: {'manage': 'showmigrations admin'}
admin
[X] 0001_initial
[X] 0002_logentry_remove_auto_add
[END] RequestId: bcc908da-3bde-11e8-8f67-bb3b239cecd Version: $LATEST
[REPORT] RequestId: bcc908da-3bde-11e8-8f67-bb3b239cecd Duration: 174.89 ms
Billed Duration: 200 ms
Memory Size: 512 MB
Max Memory Used: 81 MB

(imageGalleryProject-KAW6aJSD) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ █
```

```
(imageGalleryProject-KAW6aJSD) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ zappa invoke dev "print (1 + 2 + 3)" --raw
Calling invoke for stage dev..
[START] RequestId: 8a119b48-3be1-11e8-92c4-85e7dddb61bb Version: $LATEST
[DEBUG] 2018-04-09T10:34:10.531Z 8a119b48-3be1-11e8-92c4-85e7dddb61bb Zappa Event: {'raw_command': 'print (1 + 2 + 3)'}
6
[END] RequestId: 8a119b48-3be1-11e8-92c4-85e7dddb61bb Version: $LATEST
[REPORT] RequestId: 8a119b48-3be1-11e8-92c4-85e7dddb61bb Duration: 20.11 ms
Billed Duration: 100 ms
Memory Size: 512 MB
Max Memory Used: 80 MB

(imageGalleryProject-KAW6aJSD) abdulw@ULTP-711:~/workspace/book/chapter_5/imageGalleryProject(master)$ █
```

Chapter 7: Building a Falcon Application with Zappa



Chapter 8: Custom Domain with SSL

The screenshot shows the AWS Route 53 Dashboard. On the left sidebar, under the 'Hosted zones' section, there are several options: Hosted zones, Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains, and Pending requests. The main content area is divided into four sections: 'DNS management' (2 Hosted zones), 'Traffic management' (A visual tool for creating policies), 'Availability monitoring' (Health checks monitor applications and web resources), and 'Domain registration' (A domain is the name used to access your application). Below these sections, there's a 'Register domain' form where you can type a domain name and select a top-level domain like '.com'. There's also a 'Check' button. To the right, there's a 'More info' section with links to developer guides, FAQs, and forums. At the bottom, there's a 'Service health' section showing a green checkmark for 'Amazon Route 53'.

The screenshot shows the 'Hosted zones' page in the AWS Route 53 console. The left sidebar has the same navigation as the dashboard. The main area displays a table of existing hosted zones, including 'conguru.in' (Public, 8 records) and 'university.info' (Public, 3 records). To the right of the table, there's a 'Create Hosted Zone' form. It includes fields for 'Domain Name' (set to 'abdulwahid.info'), 'Comment' (empty), and 'Type' (set to 'Public Hosted Zone'). A descriptive text explains that a public hosted zone routes traffic on the Internet. At the bottom right of the form is a 'Create' button.

The screenshot shows the AWS Route 53 console. The left sidebar has 'Hosted zones' selected. The main area displays two record sets for the domain 'abdulwahid.info'. The first record set contains NS records for 'ns-1296.awsdns-34.org.', 'ns-1745.awsdns-26.co.uk.', 'ns-756.awsdns-30.net.', and 'ns-248.awsdns-31.com.'. The second record set contains an SOA record for 'ns-1296.awsdns-34.org.' with the value 'awsdns-hostmaster.amazon...'. A message at the bottom right says 'To get started, click Create Record Set button or click an existing record set.'

The screenshot shows the GoDaddy Domain Manager interface. In the 'Nameservers' section, it lists 'ns59.domaincontrol.com' and 'ns60.domaincontrol.com'. Below this are sections for 'Advanced Features' (Manage Templates, Export Zone File (Unix), Export Zone File (Windows), Import Zone File, DNSSEC) and 'Forwarding' (Domain: Not set up, Subdomain: Not set up). There are 'ADD' buttons for both.

The screenshot shows the 'Nameservers' section in the GoDaddy Domain Manager. It displays a list of nameservers with a 'Custom' type selected. The nameservers listed are ns-1296.awsdns-34.org, ns-1745.awsdns-26.co.uk, ns-756.awsdns-30.net, and ns-248.awsdns-31.com. Each entry has a small trash can icon to its right. At the bottom, there are 'Save' and 'Cancel' buttons, and a 'Add Nameserver' button.

The screenshot shows the 'Request a certificate' page in the AWS Certificate Manager. It starts with a note about importing an existing certificate. Below it, there's a section for requesting a new certificate. The 'Request a public certificate' option is selected. To the right, there are two bullet points: 'Request a public certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.' and 'Request a private certificate from your organization's certificate authority.' Both points have a 'Learn more' link. At the bottom right are 'Cancel' and 'Request a certificate' buttons. The footer includes links for Feedback, English (US), Privacy Policy, and Terms of Use.

The screenshot shows the AWS Certificate Manager console with the URL <https://console.aws.amazon.com/acm/home?region=us-east-1#/wizard>. The user is on Step 1: Add domain names. A sidebar on the left lists steps: Step 1: Add domain names (highlighted in orange), Step 2: Select validation method, Step 3: Review, and Step 4: Validation. The main content area has two informational boxes: one about CT logging and another about using certificates with other AWS services. Below these is a section titled "Add domain names" with a text input field containing "*abdulwahid.info". A "Remove" button is next to the input field. A "Add another name to this certificate" button is also present. A note below explains how to add additional names to the certificate. At the bottom, a message says "At least one domain name is required" and includes "Cancel" and "Next" buttons.

The screenshot shows the AWS Certificate Manager console on Step 2: Select validation method. The sidebar still lists the four steps. The main content area is titled "Select validation method". It explains that ACM validates ownership of domains by using DNS or email. Two options are shown: "DNS validation" (selected) and "Email validation". Each option has a description and a "Learn more" link. At the bottom, there are "Cancel", "Previous", and "Review" buttons.

AWS Certificate Manager

Secure | https://console.aws.amazon.com/acm/home?region=us-east-1#/wizard/

Abdul Wahid N. Virginia Support

Request a certificate

Step 1: Add domain names
Step 2: Select validation method
Step 3: Review
Step 4: Validation

Review

Review your choices.

Domain name

The name you want to secure with an SSL/TLS certificate.

Domain name *.*.abdulwahid.info

Validation method

The method AWS uses to validate your certificate request.

Validation method DNS

Cancel Previous **Confirm and request**

AWS Certificate Manager

Secure | https://console.aws.amazon.com/acm/home?region=us-east-1#/wizard/

Abdul Wahid N. Virginia Support

Request a certificate

Step 1: Add domain names
Step 2: Select validation method
Step 3: Review
Step 4: Validation

Request in progress
A certificate request with a status of Pending validation has been created. Further action is needed to complete the validation and approval of the certificate.

Validation

Create a CNAME record in the DNS configuration for each of the domains listed below. You must complete this step before AWS Certificate Manager (ACM) can issue your certificate, but you can skip this step for now by clicking Continue. To return to this step later, open the certificate request in the ACM Console.

Domain	Validation status
*.abdulwahid.info	Pending validation

Export DNS configuration to a file You can export all of the CNAME records to a file

Continue

Feedback English (US)

© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the 'Create record in Route 53' dialog box. It displays a table with one row for a CNAME record. The 'Name' column contains '_6232baf5bb7b1971371320f2823e4c6.abdulwahid.info.', the 'Type' column is 'CNAME', and the 'Value' column is '_9113cab6c1032617a7d1f0ee5e410ef4.acm-validations.aws.'. Below the table, there is a note about changing DNS configuration and a 'Create record in Route 53' button. At the bottom, there are 'Cancel' and 'Create' buttons, along with links for 'Export DNS configuration to a file' and 'Continue'.

Create record in Route 53

Below is your DNS record for domain validation. Click **Create** below to create the records in your Route 53 hosted zone

Hosted zone abdulwahid.info.

Name	Type	Value
_6232baf5bb7b1971371320f2823e4c6.abdulwahid.info.	CNAME	_9113cab6c1032617a7d1f0ee5e410ef4.acm-validations.aws.

Note: Changing the DNS configuration allows ACM to issue certificates for this domain name for as long as the DNS record exists. You can revoke permission at any time by removing the record. [Learn more.](#)

Create record in Route 53 Amazon Route 53 DNS Customers ACM can update your DNS configuration for you. [Learn more.](#)

Export DNS configuration to a file You can export all of the CNAME records to a file

Continue

The screenshot shows the confirmation page for the DNS validation step. It lists the domain '*.abdulwahid.info' with a 'Validation status' of 'Pending validation'. Below this, it provides instructions to add a CNAME record and displays the record details in a table. A note about changing DNS configuration is present, along with a 'Create record in Route 53' button and a success message in a green box. At the bottom, there are links for 'Export DNS configuration to a file' and 'Continue'.

Domain	Validation status
*.abdulwahid.info	Pending validation

Add the following CNAME record to the DNS configuration for your domain. The procedure for adding CNAME records depends on your DNS service Provider. [Learn more.](#)

Name	Type	Value
_6232baf5bb7b1971371320f2823e4c6.abdulwahid.info.	CNAME	_9113cab6c1032617a7d1f0ee5e410ef4.acm-validations.aws.

Note: Changing the DNS configuration allows ACM to issue certificates for this domain name for as long as the DNS record exists. You can revoke permission at any time by removing the record. [Learn more.](#)

Create record in Route 53 Amazon Route 53 DNS Customers ACM can update your DNS configuration for you. [Learn more.](#)

Success
The DNS record was written to your Route 53 hosted zone. It may take up to 30 minutes for the changes to propagate, and for AWS to validate the domain.

Export DNS configuration to a file You can export all of the CNAME records to a file

Continue

AWS Certificate Manager

Secure | https://console.aws.amazon.com/acm/home?region=us-east-1#/id=arn:aws:acm:us-east-1:042373950390:certificate%2Fafo... Abdul Wahid N. Virginia Support

Certificates

Request a certificate Import a certificate Actions

Viewing 1 to 3 of 3 certificates

	Name	Domain name	Additional names	Status	Type	In use?	Renewal eligibility
		*.abdulwahid.info		Issued	Amazon issued	No	Ineligible

Status

Status Issued
Detailed status The certificate was issued at 2018-05-24T10:42:04UTC

Domain	Validation status
*.abdulwahid.info	Success

Export DNS configuration to a file You can export all of the CNAME records to a file

The screenshot shows the AWS Certificate Manager console. On the left, there's a sidebar with 'Certificates' and 'Certificate Manager' selected. The main area is titled 'Certificates' and contains a message about AWS Certificate Manager logs domain names from your certificates into public certificate transparency (CT) logs when renewing certificates. Below this is a table showing one certificate entry:

	Name	Domain name	Additional names	Status	Type	In use?	Renewal eligibility
		*.abdulwahid.info		Issued	Amazon issued	No	Ineligible

Below the table, there's a 'Status' section with detailed information about the certificate's issuance. It shows the status as 'Issued' and provides a 'Detailed status' message: 'The certificate was issued at 2018-05-24T10:42:04UTC'. There's also a table showing validation status for the domain *.abdulwahid.info, which is listed as 'Success'. At the bottom, there's a link to 'Export DNS configuration to a file'.

Domain	Validation status
*.abdulwahid.info	Success

Add the following CNAME record to the DNS configuration for your domain. The procedure for adding CNAME records depends on your DNS service Provider. [Learn more.](#)

Name	Type	Value
_6232baf55bb7b1971371320f2823e4c6.abdulwahid.in fo.	CNAME	_9113cab6c1032617a7d1f0ee5e410ef4.acm-validations.aws.

Note: Changing the DNS configuration allows ACM to issue certificates for this domain name for as long as the DNS record exists. You can revoke permission at any time by removing the record. [Learn more.](#)

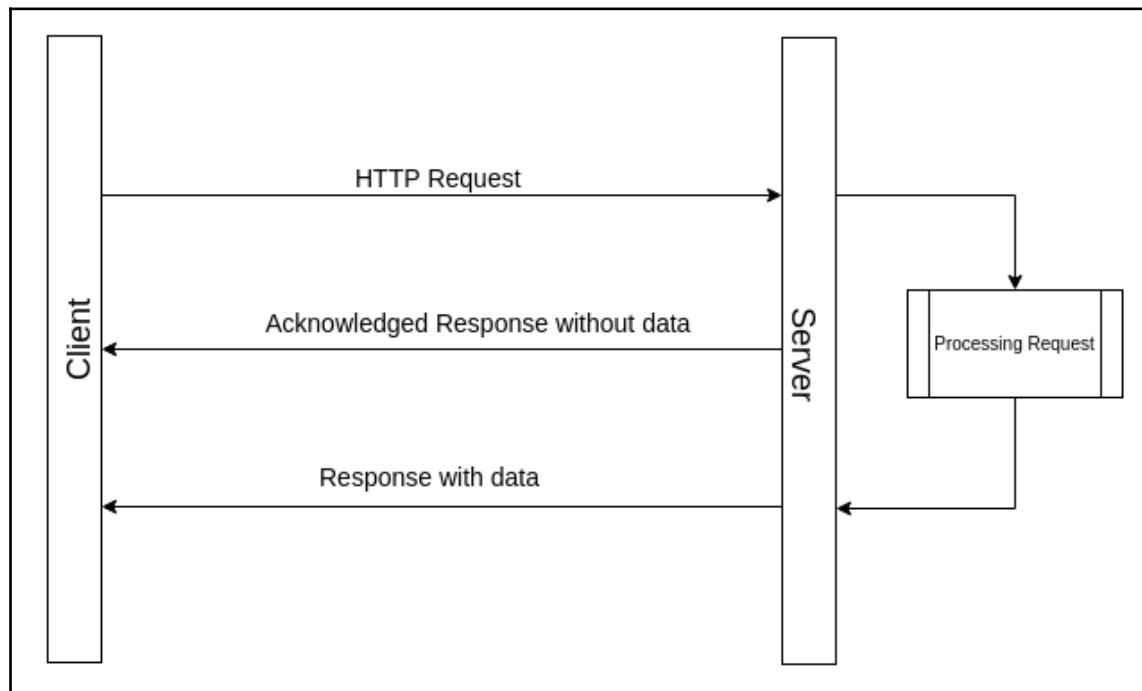
Create record in Route 53 [Amazon Route 53 DNS Customers](#) ACM can update your DNS configuration for you. [Learn more.](#)

 [Export DNS configuration to a file](#) You can export all of the CNAME records to a file

Details

Type	Amazon Issued	Requested at	2018-05-24T10:40:22UTC
In use?	No	Issued at	2018-05-24T10:42:04UTC
Domain name	*.abdulwahid.info	Not before	2018-05-24T00:00:00UTC
Number of additional names	0	Not after	2019-06-24T12:00:00UTC
Identifier	af0796fa-3a46-49ae-97d8-90a6b5ff6784	Public key info	RSA 2048-bit
Serial number	08:0b:2d:f0:06:dd:ab:73:e3:f6:0a:4c:40:d f:f9:9d	Signature algorithm	SHA256WITHRSA
		ARN	arn:aws:acm:us-east- 1:042373950390:certificate/af0796fa- 3a46-49ae-97d8-90a6b5ff6784

Chapter 9: Asynchronous Task Execution on AWS Lambda



The screenshot shows the AWS SNS Topics page. On the left sidebar, under the 'Text messaging (SMS)' section, the 'Topics' item is selected. The main content area is titled 'Topics' and contains a table with one row. The table has two columns: 'Name' and 'ARN'. The single entry is 'DailyQuoteSubscription' with the ARN 'arn:aws:sns:us-east-1:042373950390:DailyQuoteSubscription'. Below the table, it says 'Total Items: 1' and 'Selected Items: 0'. At the top of the page, there are buttons for 'Publish to topic', 'Create new topic', and 'Actions'. A 'Filter' input field is also present.

The screenshot shows the 'Text messaging preferences' page. On the left sidebar, under the 'Text messaging (SMS)' section, the 'Text messaging preferences' item is selected. The main content area is titled 'Text messaging preferences' and contains several configuration fields:

- 'Default message type': Set to 'Transactional'.
- 'Account spend limit': Set to '1 USD'.
- 'IAM role for CloudWatch Logs access': Set to 'arn:aws:iam::042373950390:role/SNSSuccessFeedback' with options to 'Change' or 'Clear'.
- 'Default percentage of success to sample': Set to '100'.
- 'Default sender ID': Set to 'QUOTE'.
- 'Reports storage': Set to 'Name of an S3 bucket'.

At the bottom right, there are 'Cancel' and 'Update preferences' buttons. The footer includes standard AWS links: Feedback, English (US), Privacy Policy, and Terms of Use.

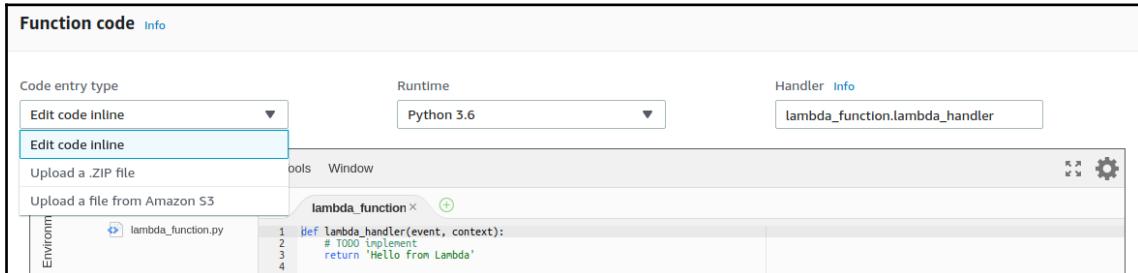
The screenshot shows the AWS SNS Subscriptions page. The left sidebar includes links for SNS dashboard, Topics, Applications, Subscriptions (which is selected and highlighted in orange), and Text messaging (SMS). The main content area has a title 'Subscriptions' with three buttons: 'Create subscription', 'Request confirmations', and 'Actions'. A 'Filter' input field is present. A table lists one subscription:

Subscription ARN	Proto...	Endpoint	Topic ARN
arn:aws:sns:us-east-1:042373950390:DailyQuoteSubscription:c455205f-59bf-4973-85ef-12...	sms	+919028790411	arn:aws:sns:us-east-1:0...

Below the table, it says 'Total Items: 1' and 'Selected Items: 0'. The bottom navigation bar includes 'Feedback', 'English (US)', a copyright notice ('© 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.'), 'Privacy Policy', and 'Terms of Use'.

This screenshot is identical to the one above, showing the AWS SNS Subscriptions page. The left sidebar and main content area are the same, including the 'Subscriptions' link which is selected. The table below the header is empty, indicating 'Total Items: 0' and 'Selected Items: 0'. The bottom navigation bar is also identical.

Chapter 10: Advanced Zappa Settings



```
abdulw@ULTP-711:~/workspace/book/Chapter09(master)$ zappa
certify      --help      package      status      unschedule
--color       init       rollback     tail       update
deploy       invoke     schedule    template   -v
-h          manage     shell       undeploy  --version
```

Chapter 11: Securing Serverless Applications with Zappa

The screenshot shows the AWS API Gateway Stage Editor for the `dev` stage of the `chapter11-dev` API. The left sidebar lists other stages like `flask-todo-dev`, `homejoy-backend-prod`, and `imagegallerypro-dev`. The main pane displays the `dev Stage Editor` with tabs for `Settings`, `Logs`, `Stage Variables`, `SDK Generation`, `Export`, `Deployment History`, and `Documentation History`. The `Settings` tab is active. It includes sections for `Cache Settings` (with an `Enable API cache` checkbox), `Default Method Throttling` (with `Enable throttling` checked, `Rate` set to 10000 requests per second, and `Burst` set to 5000 requests), `Client Certificate` (set to `None`), and `Tags` (an empty table with a `Add Stage Tag` button). A `Save Changes` button is at the bottom right.

The screenshot shows the AWS API Gateway interface. In the top navigation bar, the user is at the 'API Keys > Create' page under the 'Resource Groups' section. On the left sidebar, there's a list of APIs: chapter-3-dev, chapter-4-dev, chapter-8-dev, chapter-9-dev, flask-todo-dev, homejoy-backend-prod, and imagegallerypro-dev. Below this, the 'Usage Plans' section is visible, with 'API Keys' currently selected. The main content area is titled 'Create API Key' and contains fields for 'Name*' (Chapter11_APIKEY), 'API key*' (radio button for 'Auto Generate'), and a 'Description' field. A 'Save' button is located in the bottom right corner. A note at the bottom indicates that the 'Name' field is required.

The screenshot shows the AWS API Gateway interface, specifically the details for a created API key. The user is at the 'API Keys > dev_laqdydyrg3' page under the 'Resource Groups' section. The left sidebar shows the same list of APIs as the previous screen. The main content area displays the details of the 'dev_laqdydyrg3' API key, including its ID (zp0snz9tik), name (dev_laqdydyrg3), API key (yEddw9WeMH2UIZXHcaHQb1WvbindovrB55Rf4eAW), description (Api Key for laqdydyrg3), and status (Enabled). There is a 'Delete API Key' button in the top right. Below this, the 'Associated Usage Plans' section is shown, with a table listing the 'Chapter 11 Basic Usage Plan' associated with the 'chapter11-dev' API and stage 'dev'. An 'Edit' button is available for this association. The bottom of the screen includes standard AWS footer links for Feedback, English (US), Privacy Policy, and Terms of Use.

The screenshot shows the 'Create Usage Plan' page in the AWS API Gateway console. The left sidebar lists APIs and Usage Plans. The main area shows fields for creating a new usage plan:

- Name***: Chapter 11 Basic Usage Plan
- Description**: (empty)
- Throttling**:
 - Enable throttling**: checked
 - Rate***: 10 requests per second
 - Burst***: 20 requests
- Quota**:
 - Enable quota**: checked
 - 100 requests per Month**

* Required

Next

Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the details of the 'Chapter 11 Basic Usage Plan' (ID: 9bj515). The left sidebar lists APIs and Usage Plans. The main area displays the usage plan details:

- ID**: 9bj515
- Name**: Chapter 11 Basic Usage Plan
- Description**: No description.
- Rate**: 10 requests per second
- Burst**: 20 requests
- Quota**: 100 requests per month starting on the 1st day

Associated API Stages

Add API Stage

API	Stage	Method Throttling	Configure Method Throttling
chapter11-dev	dev	No Methods Configured	Configure Method Throttling

Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

AWS Notification Message ▶ Inbox Updates

AWS Notifications no-reply@sns.amazonaws.com via amazones.com
to me ▾

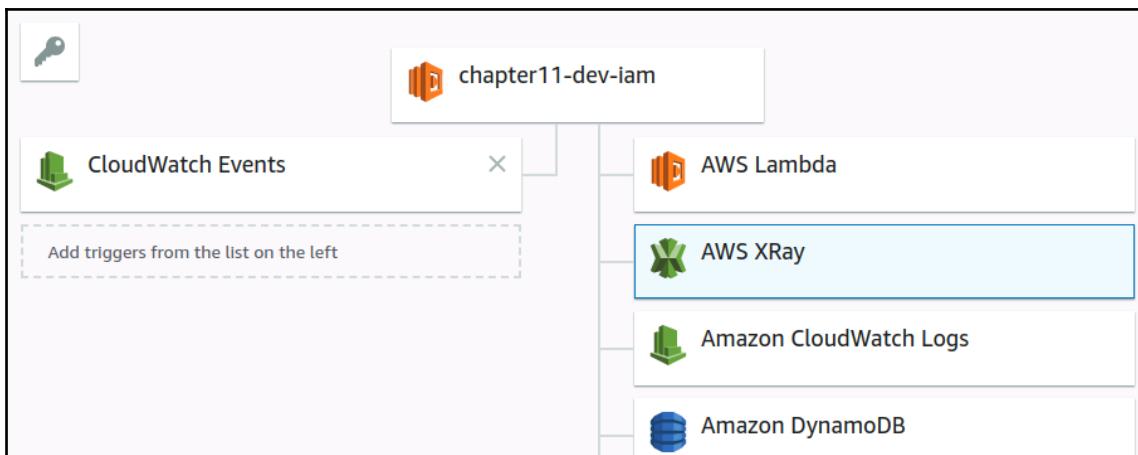
4:20 AM (31 minutes ago) Star Reply Print Forward

{"task_path": "resources.async_task", "capture_response": false, "response_id": null, "args": [], "kwargs": {}, "command": "zappa.async.route_lambda_task"}

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
<https://sns.ap-south-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-south-1:042373950390:UnhandledException:2b084dd2-a692-40c1-b09b-a9bad7a9dac8&Endpoint=abdulwahid24@gmail.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>



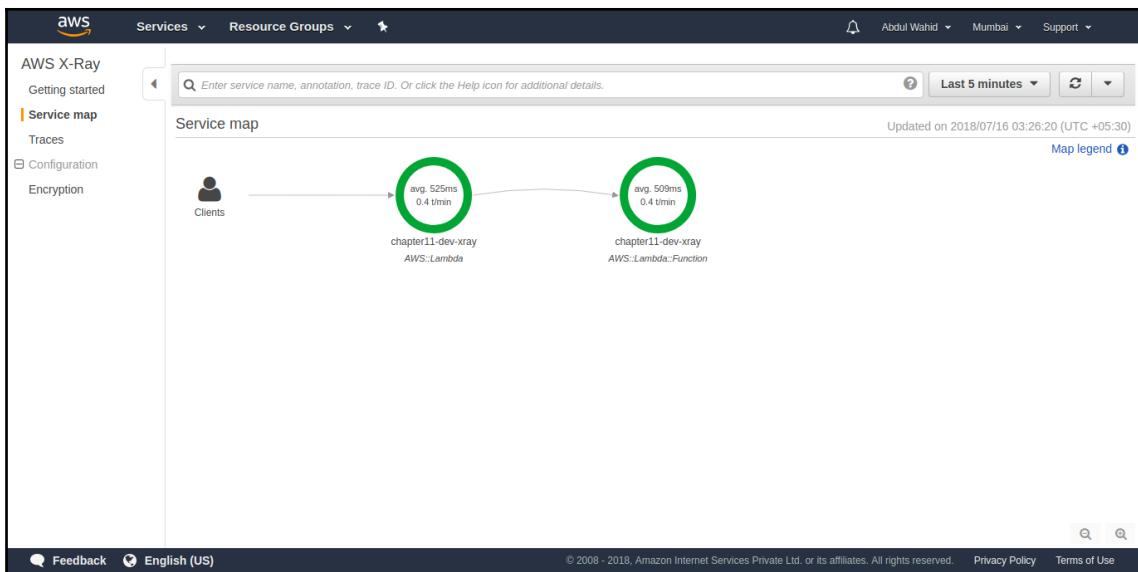
AWS XRay

Below is a summary of the AWS XRay resources and API calls your function's code has access to, as configured by its execution role. Visit the [IAM console](#) to manage these permissions.

[By action](#) [By resource](#)

Resource	Actions
All resources	Allow: xray:PutTraceSegments Allow: xray:PutTelemetryRecords

ⓘ This information was sourced from the following policy statements:
Inline policy zappa-permissions, statement 2



Network

VPC Info

Select a VPC that your function will access.

Default **vpc-4730872e (172.31.0.0/16)**



Subnets*

Select the VPC Subnets that Lambda should use to set up your VPC configuration. Format: "subnet-id (cidr-block) | az name-tag".



A Please select at least 1 subnet.

Security Groups*

Select the VPC Security Groups that Lambda should use to set up your VPC configuration. Format: "sg-id (sg-name) | name-tag". The table below will show the inbound and outbound rules for the security groups you have selected.



A Please select at least 1 security group.

- i** When you enable VPC, your Lambda function will lose default internet access. If you require external internet access for your function, ensure that your security group allows outbound connections and that your VPC has a NAT gateway.

Inbound rules

Outbound rules

< 1 >

Security group ID	Ports	Source
None	-	-

The screenshot shows the AWS VPC Dashboard. On the left, a sidebar lists various VPC-related services: Virtual Private Cloud, Your VPCs, Subnets (which is selected), Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, Security Groups, VPN Connections, Customer Gateways, Virtual Private Gateways, and VPN Connections. At the top, there are tabs for 'Create subnet' and 'Actions'. A search bar allows filtering by tags and attributes or searching by keyword. Below the search bar is a table with the following data:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Route table
subnet-1b10a072	vpc-4730872e	available	172.31.16.0/20	4091	-	ap-south-1a	rtb-29229e40	
subnet-6303f22e	vpc-4730872e	available	172.31.0.0/20	4089	-	ap-south-1b	rtb-29229e40	

At the bottom of the dashboard, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Security Groups' section, the 'default' security group is selected. The main content area displays the 'Create Security Group' and 'Security Group Actions' tabs. A search bar at the top right shows 'default'. Below it is a table with one row, showing the details of the 'default' security group. The table columns are Name tag, Group ID, Group Name, VPC, and Description. The group has a Name tag 'sg-892c4be0', Group ID 'sg-892c4be0', Group Name 'default', VPC 'vpc-4730872e', and Description 'default VPC security group'. Below the table, there is a summary card for the security group 'sg-892c4be0'. It shows the Group name: default, Group ID: sg-892c4be0, VPC: vpc-4730872e, and Group description: default VPC security group. There are tabs for Summary, Inbound Rules, Outbound Rules, and Tags, with 'Summary' being the active tab. At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

Name tag	Group ID	Group Name	VPC	Description
sg-892c4be0	sg-892c4be0	default	vpc-4730872e	default VPC security group

sg-892c4be0

Group name: default
Group ID: sg-892c4be0
VPC: vpc-4730872e
Group description: default VPC security group

Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Chapter 12: Zappa with Docker

The screenshot shows a request configuration window for a POST method to the URL `http://localhost:8000/doc-parser`. The body content type is set to `multipart/form-data`. A file named `sample-doc-file.doc` (23040 bytes) is attached under the field name `file`. The response status is `200 OK` with a duration of `10.10 ms`. The response body contains the JSON output:

```
{ "text": "This is a sample text for demo.", "filename": "sample-doc-file.doc" }
```