



CITS5503 AWS API Gateway

Dr Zhi Zhang

Overview

- API Gateway
- Final Exam

What is API Gateway?

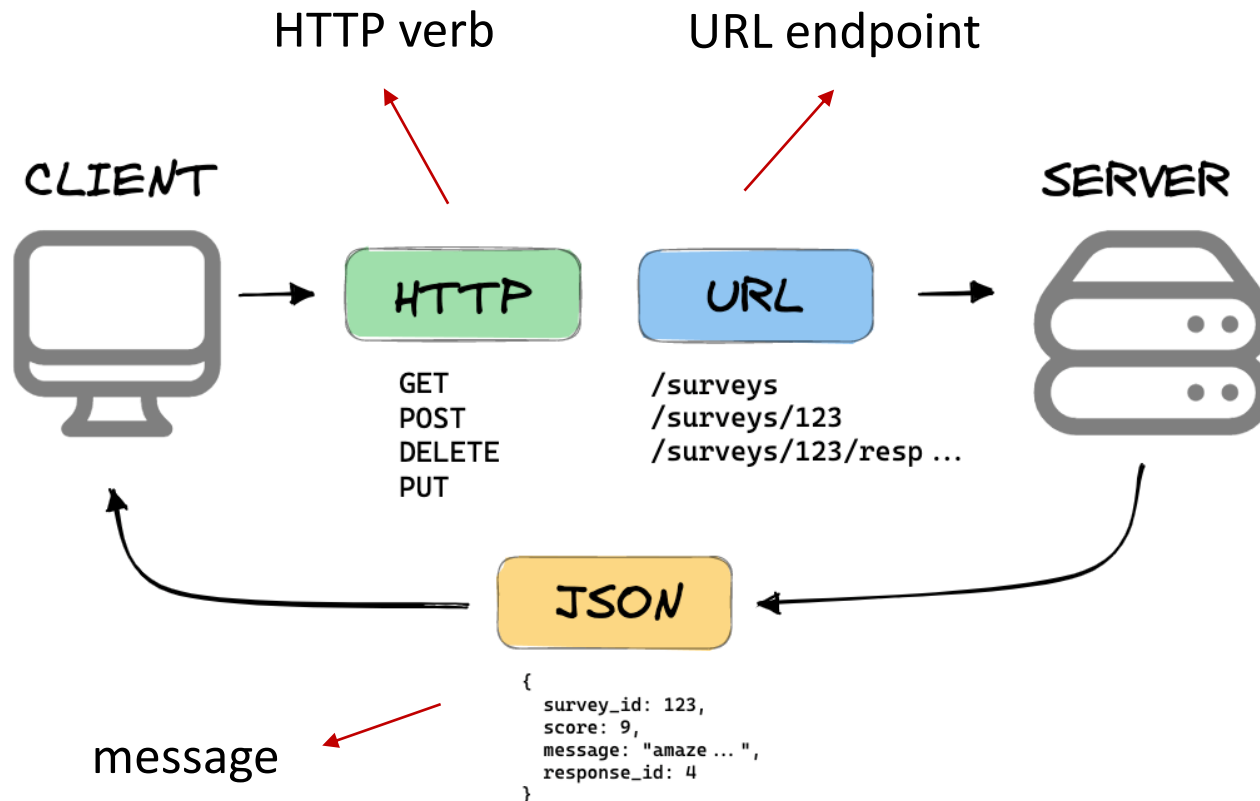
- It is an AWS service for creating, publishing, maintaining web APIs at any scale.
- What is API?

What is API Gateway?

- It is a service for creating, publishing, maintaining, monitoring, and securing web APIs at any scale.
- What is API?
 - An application programming interface defines the rules for a client or an application to communicate with a target application.
 - e.g., a timesheet application exposes an API that requests for a student's ID and a range of UWA semesters as inputs.
 - **Web API:** API used for communication on the web
 - **Examples:** REST/RESTful API, WebSocket API

What is REST/RESTful API?

- Representational State Transfer (REST) API is a software architecture that manages http(s)-based communications over the web.



HTTP verbs

GET: make a read only request to view a resource from the server

POST: create a new resource specified in the request

DELETE: destroy a specified resource on the server

PUT: update the resource based on the request or create a new one if not exist

PATCH: update the resource if it exists

URL endpoints

- A URL endpoint represents any given resource one HTTP verb can access
 - e.g., `example.com/surveys`, `example.com/surveys/123/responses`

Common HTTP status codes

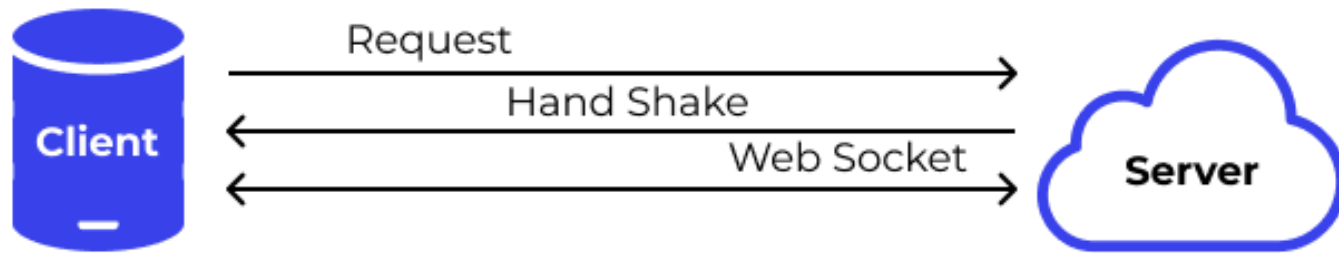
Status code	Meaning
200 OK	A successful request.
401 Unauthorized	Server requires authentication.
403 Forbidden	Client authenticated but does not have permissions to view certain resources.
404 Not Found	Page not found because no search results or may be out of stock.
500 Internal Server Error	Server side error.
503 Server Unavailable	Server side error.

- REST API: employs a request/response model where a client sends a request to a web service and the service responds back synchronously.
- WebSocket API: employs a bidirectional model: clients can send messages to a web service, and the service can also independently send messages to clients.

What is WebSocket?

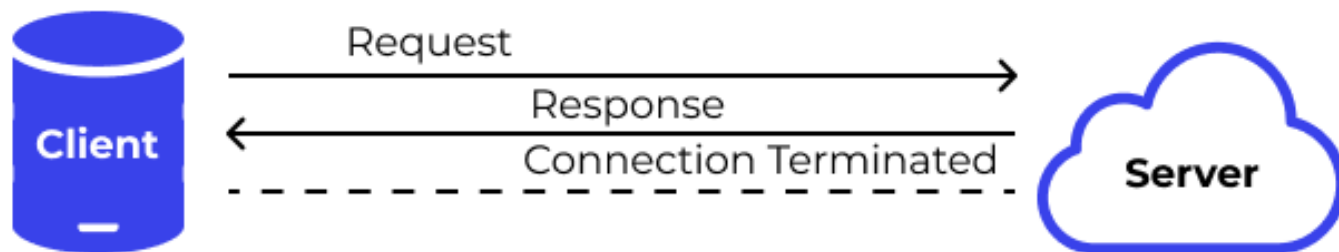
- WebSocket: a bidirectional, full-duplex protocol that is also used in the web scenarios.
- The differences between HTTP(S) and WebSocket?
 - Minor: HTTP(S) request starts with http(s)://, WebSocket request starts with ws(s)://
 - Major: While HTTP(S) is a stateless protocol, WebSocket is stateful.

WebSocket Connection

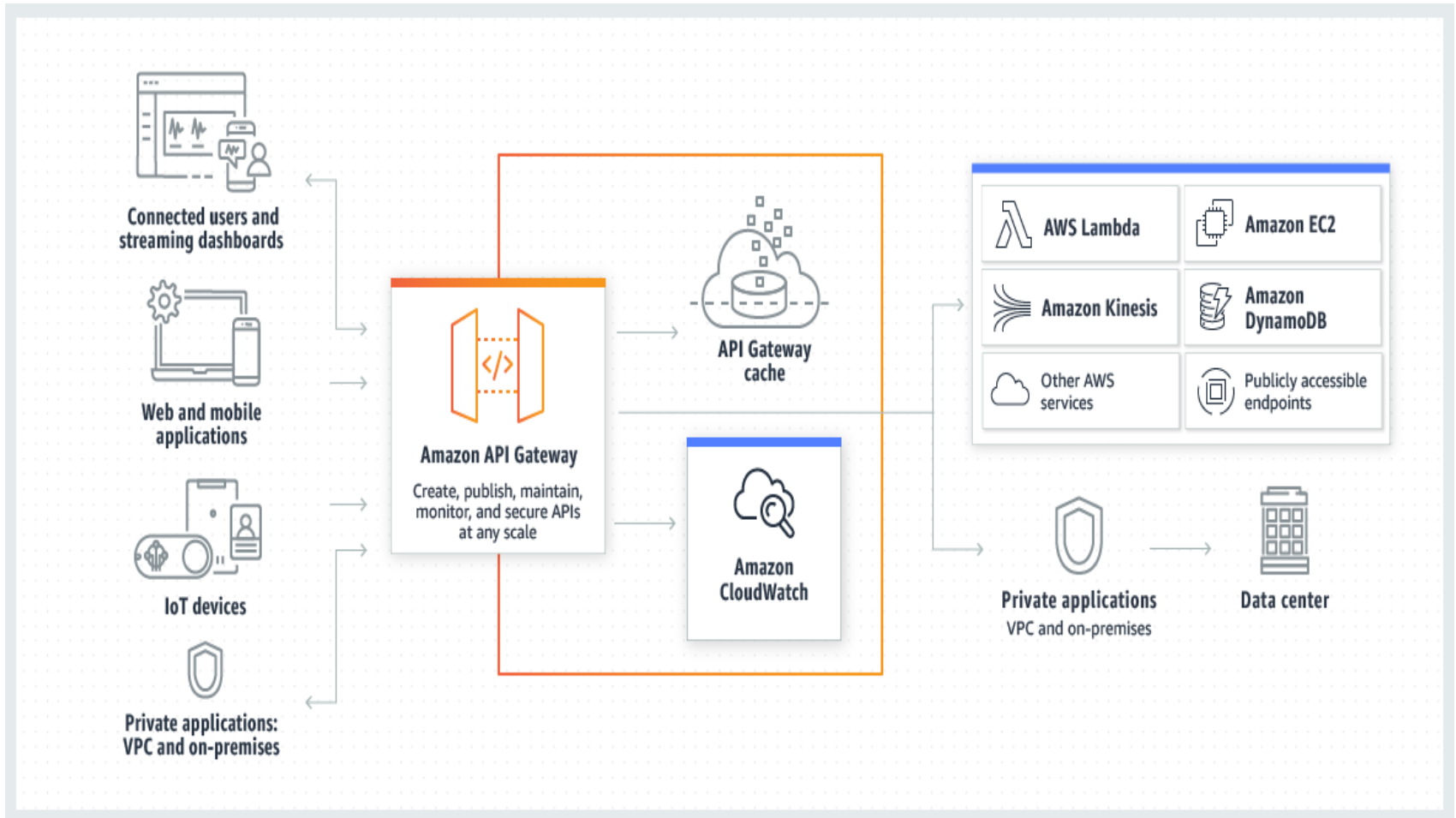


VS

HTTP Connection



How AWS API Gateway works

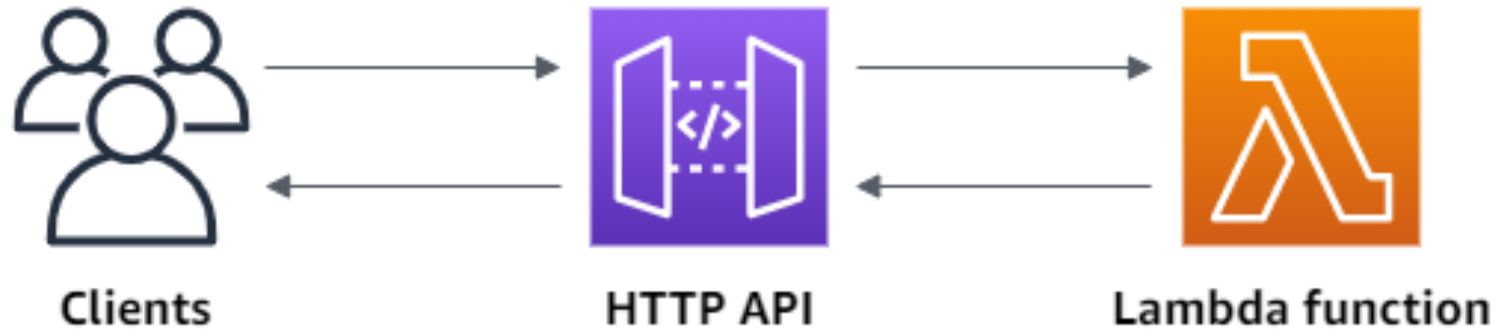


- API Gateway supports REST APIs, **HTTP APIs**, **WebSocket APIs**.

An example: HTTP API

An example: WebSocket API

An example: HTTP API



Step1: create a Lambda function

- ☒ Author from scratch
Start with a simple Hello World example.

Basic information

Function name

Enter a name that describes the purpose of your function.

lambda_httpAPI

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)

Choose the language to use to write your function. Note that the console

Node.js 18.x

Architecture [Info](#)

Choose the instruction set architecture you want for your function code.

- ☒ x86_64
- ☐ arm64

Step1: create a Lambda function



The screenshot displays the AWS Lambda console's 'Code source' page. At the top, there are tabs for 'Code source' and 'Info'. Below this is a menu bar with options: File, Edit, Find, View, Go, Tools, Window, a blue 'Test' button, and a 'Deploy' button. A search bar labeled 'Go to Anything (Ctrl-P)' is positioned above the file explorer. The file explorer on the left shows a folder named 'lambda_httpAPI' containing a file 'index.mjs'. The main editor area shows the content of 'index.mjs' with the following code:

```
1 export const handler = async (event) => {
2
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify('Hello from Lambda!'),
6   };
7   return response;
8 };
9
```

- When the handler lambda function is invoked, it returns a status code of 200 and a JSON message of Hello from Lambda.

Step2: create an HTTP API

API Gateway ×

APIs

Custom domain names

VPC links

Use the new console

As of 30th October 2023 the old console will no longer be available.

Choose an API type

HTTP API

Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

Works with the following:
Lambda, HTTP backends

Import

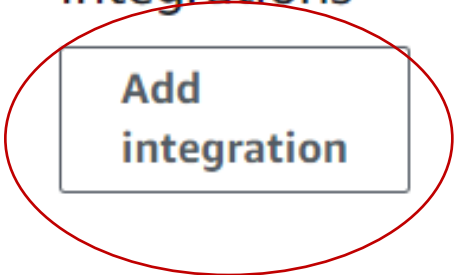
Build

Step2.1: create and configure integrations

Create and configure integrations

Specify the backend services that your API will communicate with. These are called integrations. For a Lambda integration, API Gateway invokes the Lambda function and responds with the response from the function. For HTTP integration, API Gateway sends the request to the URL that you specify and returns the response from the URL.

Integrations



**Add
integration**

Step2.1: create and configure integrations

Integrations

Lambda



Remove

AWS Region

Lambda function

Version [Learn](#)

ap-nor... ▼



arn:aws:lambda:ap-northeast-1: X

[more.](#)

2.0



Add
integration

Step2.2: specify the API name

API name

An HTTP API must have a name. This name is cosmetic and does not have to be unique; you will use the API's ID (generated later) to programmatically refer to this API.

Cancel

Review and Create

Next

Step2.3: create routes

API Gateway uses routes to expose integrations to consumers of your API.
Routes for HTTP APIs consist of two parts: an HTTP method and a resource path (e.g., GET /pets). You can define specific HTTP methods for your integration (GET, POST, PUT, PATCH, HEAD, OPTIONS, and DELETE) or use the ANY method to match all methods that you haven't defined on a given resource.

Method	Resource path	Integration target	
ANY ▼	/lambda_httpAPI	lambda_httpAPI ▼	Remove
→			
<div>Add route</div>			

Step3: test the HTTP API

Stages for example_httpAPI			
<input type="text" value="Find resources"/>			
Stage name	Invoke URL	Attached deployment	Auto deploy
\$default	https://7j8542940g.execute-api.ap-northeast-1.amazonaws.com	gsyg7p	enabled

- Full URL:
https://7j8542940g.execute-api.ap-northeast-1.amazonaws.com/lambda_httpAPI

Step3: test the HTTP API

Stages for example_httpAPI

Find resources

Stage name	Invoke URL	Attached deployment	Auto deploy
\$default	https://7j8542940g.execute-api.ap-northeast-1.amazonaws.com	gsyg7p	enabled

← → 🔒 7j8542940g.execute-api.ap-northeast-1.amazonaws.com/lambda_httpAPI

"Hello from Lambda!"

Step4: clean up

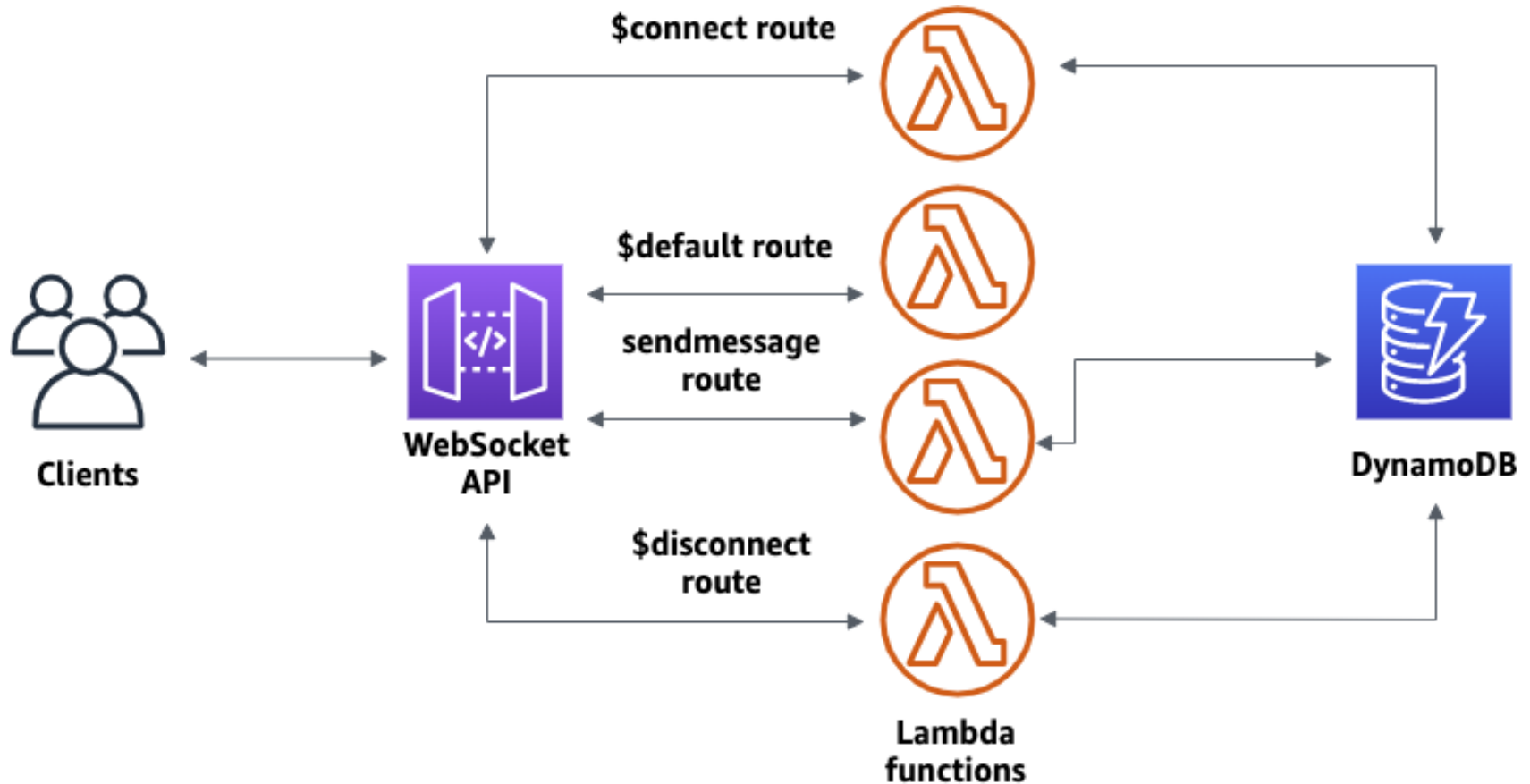
- delete the HTTP API.
- delete the Lambda function, its log group, and execution role.

- API Gateway supports REST APIs, **HTTP APIs**, **WebSocket API**.

An example: HTTP API

An example: **WebSocket API**

An example: WebSocket API

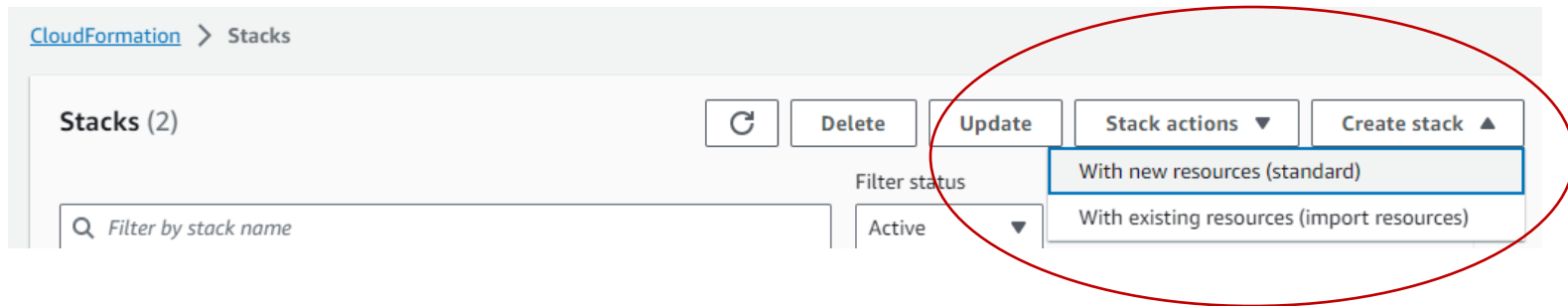


Step1: create Lambda functions and a DynamoDB table via CloudFormation

- CloudFormation: a service that helps us model and set up the AWS resources we need.
 - A template is used to describe all the AWS resources that are needed.

Step1.1: create Lambda functions and a DynamoDB table via CloudFormation

- CloudFormation: a service that helps us model and set up the AWS resources we need.



Step1.2: prepare template

- template: a JSON or YAML file that contains configuration information about the AWS resources we need.

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready

☐ Use a sample template

☐ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL

☒ Upload a template file

Upload a template file

 Choose file

JSON or YAML formatted file

<https://docs.aws.amazon.com/apigateway/latest/developer-guide/samples/ws-chat-app-starter.zip>

Step1.2: prepare template

- An example template that creates a DynamoDB table and Lambda functions.

```
Resources:
  ConnectionsTable8000B8A1:
    Type: AWS::DynamoDB::Table
    Properties:
      KeySchema:
        - AttributeName: connectionId
          KeyType: HASH
      AttributeDefinitions:
        - AttributeName: connectionId
          AttributeType: S
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 5
      UpdateReplacePolicy: Delete
      DeletionPolicy: Delete
```

Step1.3: specify stack details

Specify stack details

Stack name

Stack name

example-WebSocketAPI

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Capabilities



The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. [Learn more](#)



I acknowledge that AWS CloudFormation might create IAM resources.

Create change set

Cancel

Previous

Submit

Step2: create a WebSocket API

API Gateway ×

APIs

Custom domain names

VPC links

Use the new console

As of 30th October 2023 the old console will no longer be available.

Works with the following:
Lambda, HTTP backends

Import

Build

WebSocket API

Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.

Works with the following:
Lambda, HTTP, AWS Services

Build

Step2.1: specify API details

Specify API details

API name

API name

A unique ID will also be generated, and it can be used to programmatically refer to this API.

The name is cosmetic and does not have to be unique.

Step2.2: add predefined routes

Add routes

API Gateway uses routes to expose integrations to clients. API Gateway evaluates the route selection expression of your API at runtime to determine which route to invoke.

Predefined routes

The \$connect route is triggered when a client connects to your API.

Route key

\$connect

Remove

The \$disconnect route is triggered when either the server or the client closes the connection.

Route key

\$disconnect

Remove

The \$default route is triggered if the route selection expression can't be evaluated against the message or if no matching route is found.

Route key

\$default

Remove

Step2.3: add custom routes

Custom routes

Add custom routes to invoke integrations based on message content.

Route key

Remove

Add custom route

Cancel

Previous

Next

Step2.4: attach integrations

Attach integrations

To deploy this API, you must set up at least one route. All routes that you set up must have an integration attached. To set up integrations later, select Mock as the integration type for your routes.

Integration for \$connect

Integration type

Lambda

AWS Region

ap-south... ▼

Lambda function

🔍 arn:aws:lambda:ap-southeast-2:489389878001: ✕

Step2.4: attach integrations

Integration for \$disconnect

Integration type

Lambda

AWS Region

ap-south... ▼

Lambda function

arn:aws:lambda:ap-southeast-2:489389878001: X

Integration for \$default

Integration type

Lambda

AWS Region

ap-south... ▼

Lambda function

arn:aws:lambda:ap-southeast-2:489389878001: X

Integration for sendmessage

Integration type

Lambda

AWS Region

ap-south... ▼

Lambda function

arn:aws:lambda:ap-southeast-2:489389878001: X

Cancel

Previous

Next

Step3: test the WebSocket API

WebSocket URL: <wss://wcdyw35fn2.execute-api.ap-southeast-2.amazonaws.com/production>
Connection URL: <https://wcdyw35fn2.execute-api.ap-southeast-2.amazonaws.com/production/@connections>

- wscat

```
sudo apt install nodejs  
sudo apt install npm  
sudo npm install -g wscat
```

Step3.1: demo

- Demo

- connect to the API

```
wscat -c wss://wcdyw35fn2.execute-api.ap-southeast-2.amazonaws.com/production
```

- send a message

```
{"action": "sendmessage", "message": "hello, CITS5503!"}
```

- disconnect from the API

```
CTRL+C
```

Exam Format

- The exam is F2F and has a time limit of 2 hours
- The exam is marked out of 100 points and is worth 50% of the overall unit mark.
- The exam is open book: any printed or written materials are allowed (no page limit).
- UWA approved calculators with stickers are needed.

Exam Instructions

- You will be provided with an answer booklet
- You should clearly state which questions of the final exam you are answering.
- Please do NOT use a pencil or a red pen.
- Please use readable handwriting. What cannot be read cannot be marked.

Exam Topics

- 7 topics within lectures and labs (excluding optional steps)
 - Networking
 - IAM
 - AI
 - DevOps
 - Web Application Architecture and Design
 - Storage
 - Cloud computing introduction
- More than 90% of the questions are from the first 5 topics.

Question Format

- 7 themed questions in total and each themed question describes a scenario, based on which, some sub-questions are asked.
- The format the sub-question is the same as that of part C in the mid-sem test, i.e., short answer questions.
- A few sub-questions were picked from the assignments.md of relevant topics, excluding [13][14] in Week 5, [17] in Week 8, [18] in Week 9, [20], [21] in Week 10 and [22] in Week 11.

To prepare for the exam

- Carefully review relevant slides (recordings), labs (excluding optional steps), and assignments.md.