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?

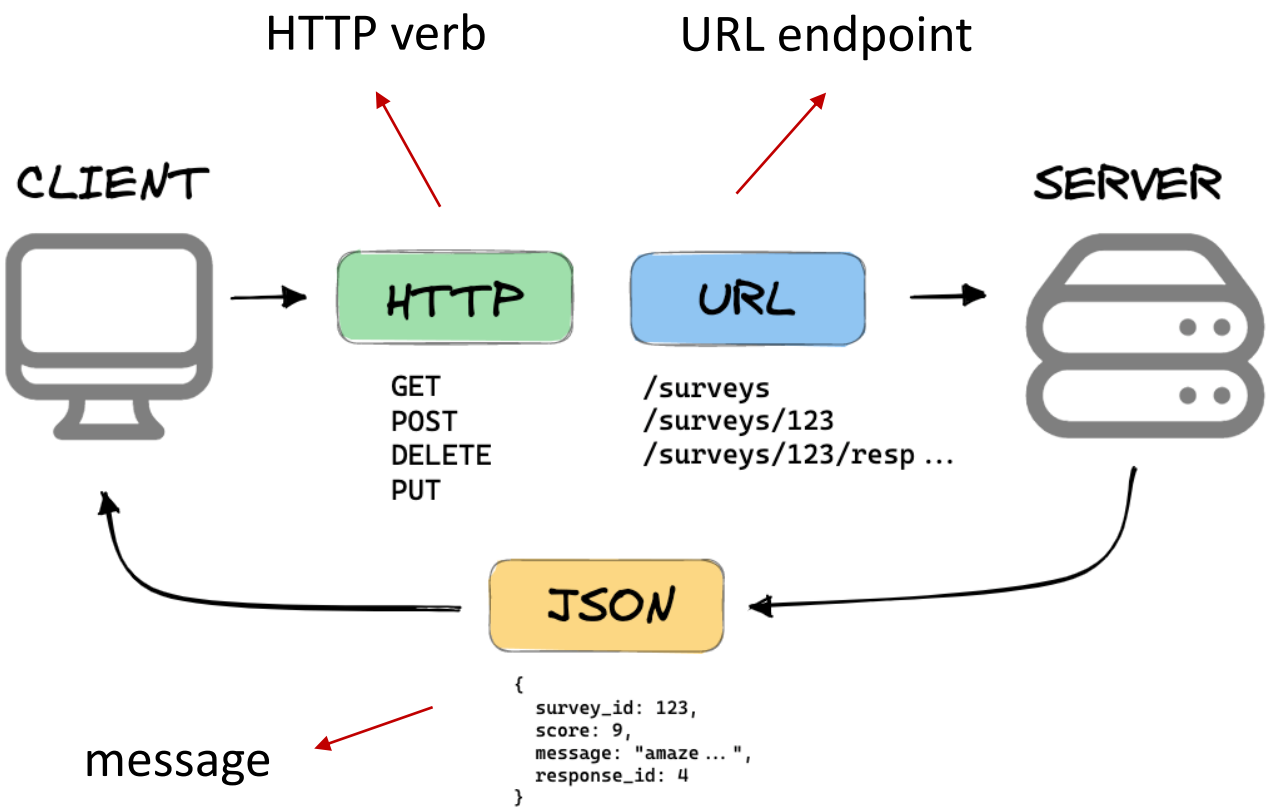
What is API Gateway?

- 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.



https://mannhowie.com/rest-api

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

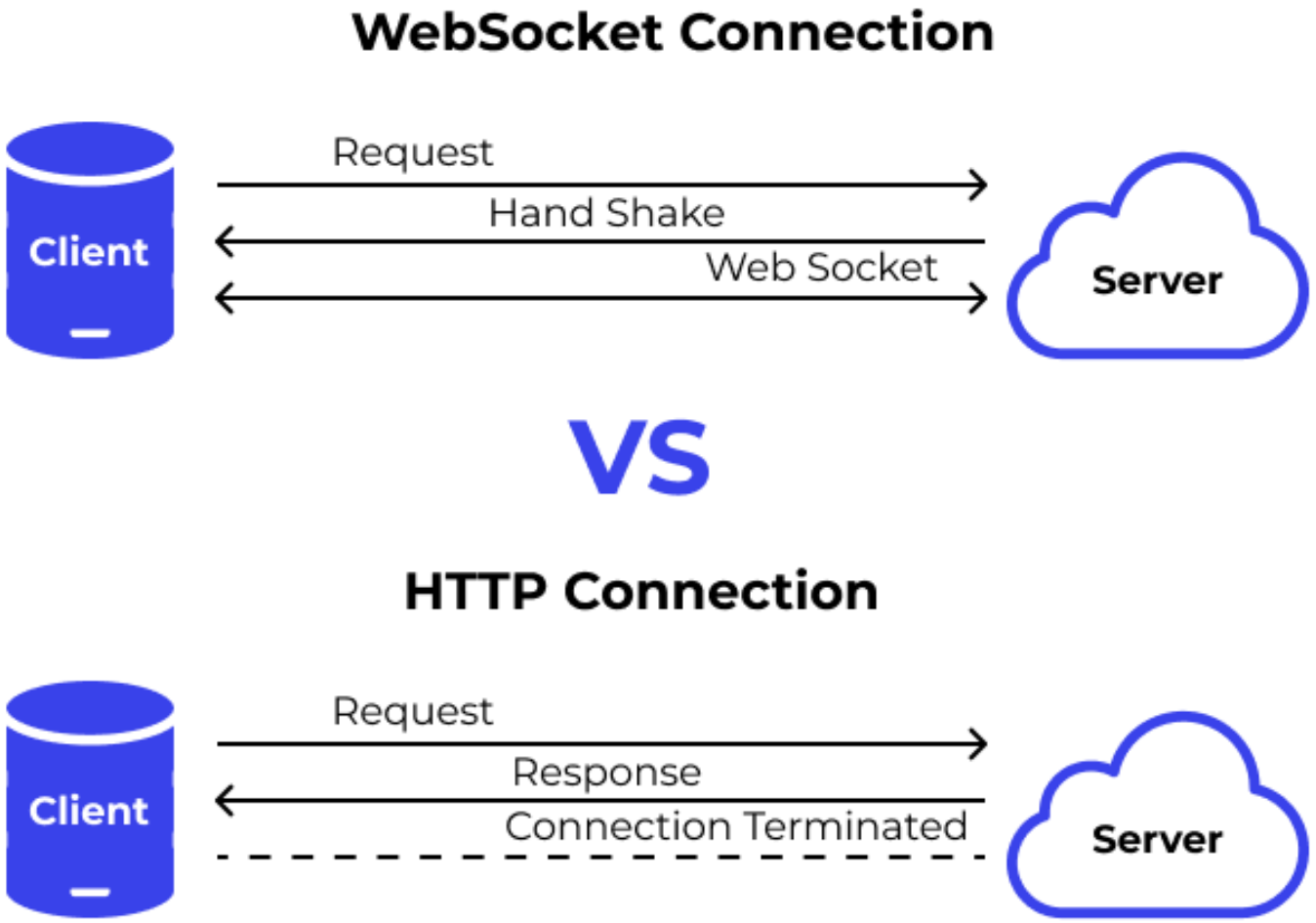
Status code	**Meaning**
200 OK	A successful request.
401 Unauthorized	Server requires authentication.
403 Forbidden	Client authenticated but does not have permissions to view certain resources.
404 Not Found	Page not found because no search results or may be out of stock.
500 Internal Server Error	Server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

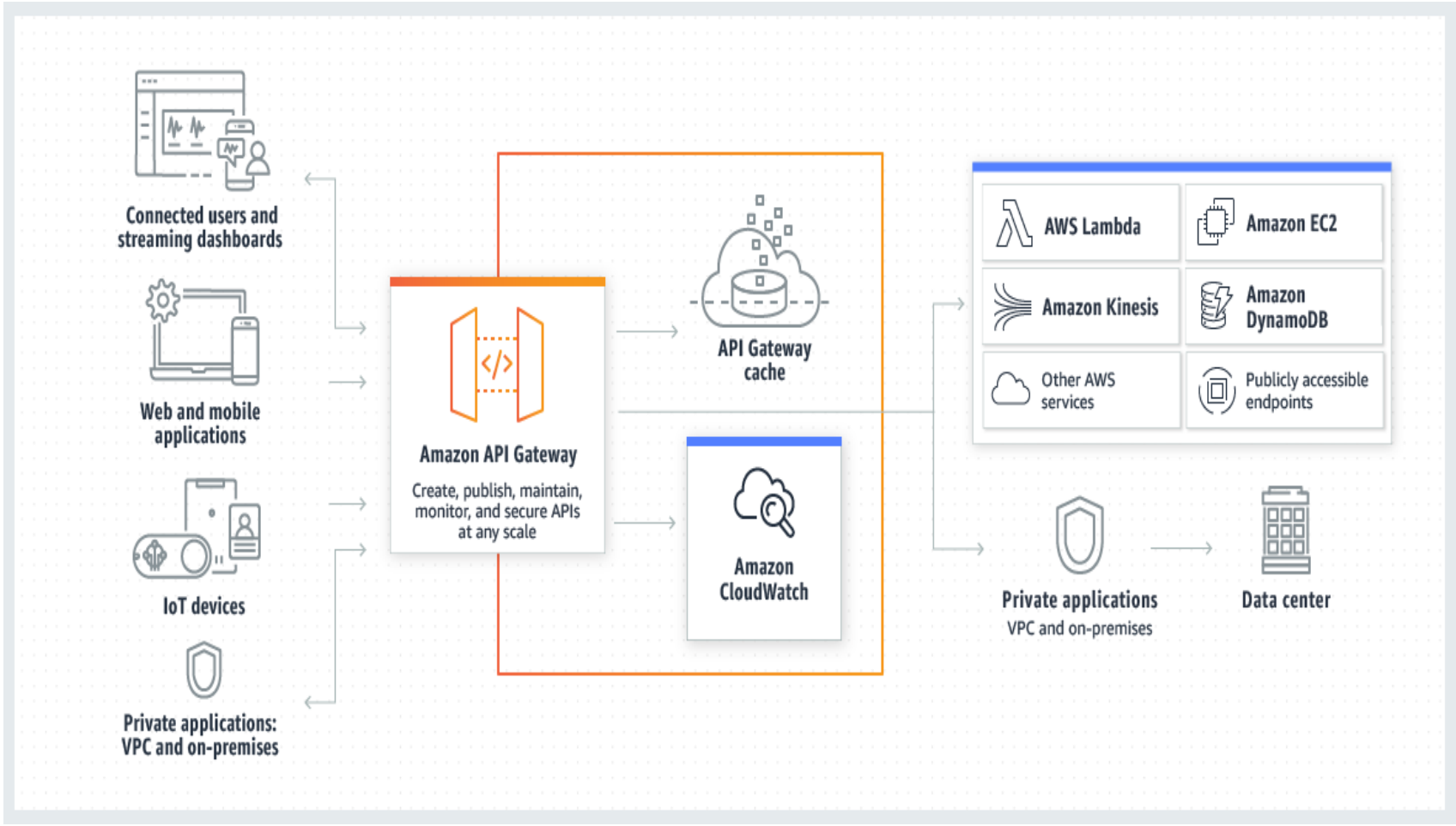
What is WebSocket?

- 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.



https://www.wallarm.com/what/a-simple-explanation-of-what-a-websocket-is

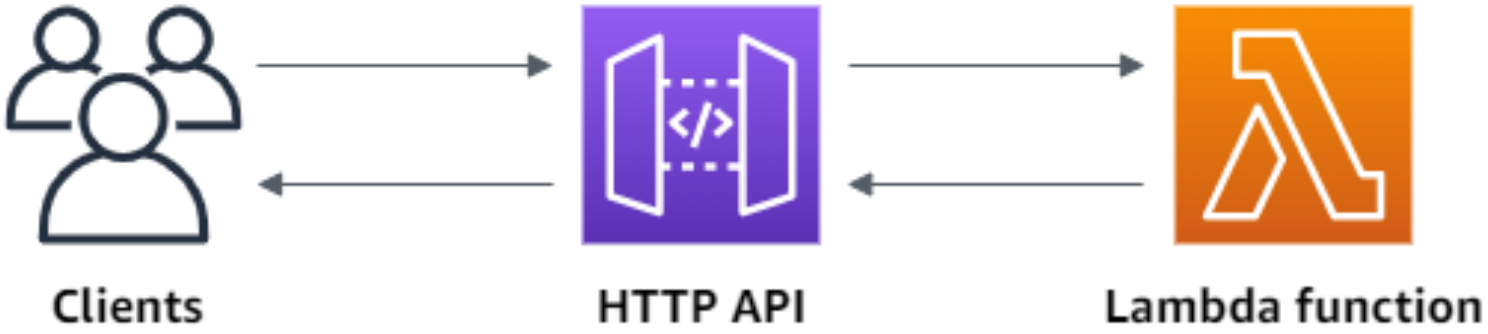
How AWS API Gateway works



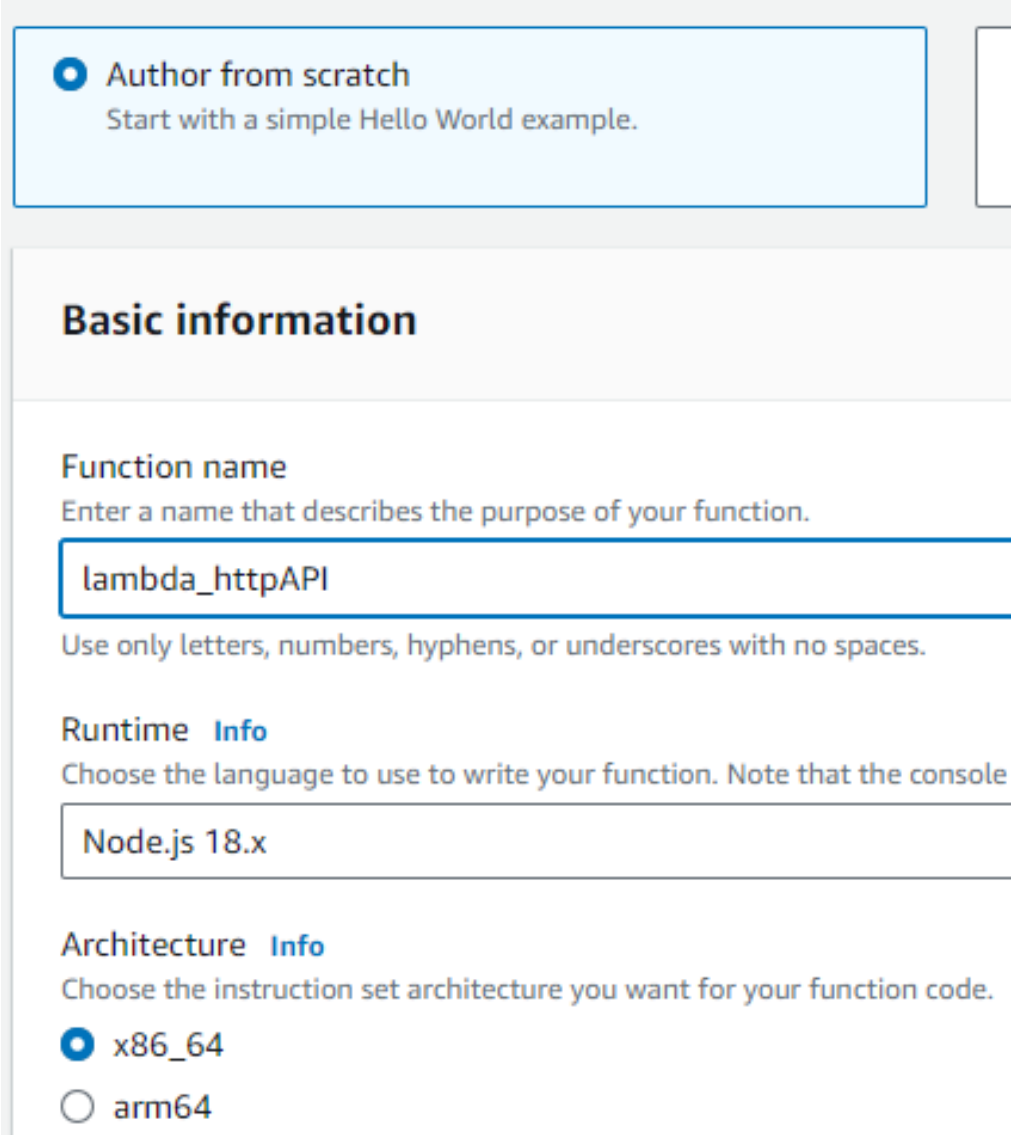
- API Gateway supports REST APIs, **HTTP APIs, WebSocket APIs.**

An example: HTTP API
An example: WebSocket API

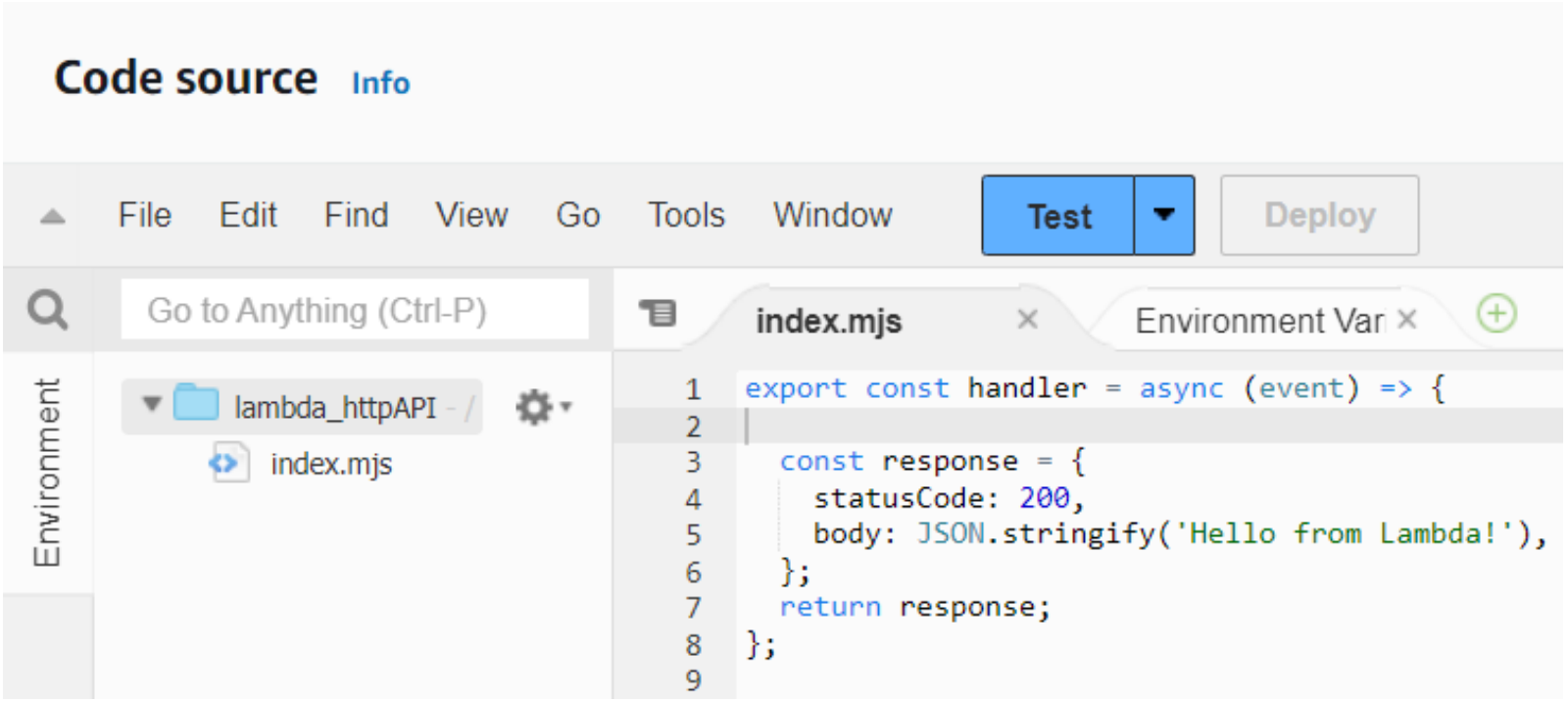
An example: HTTP API



Step1: create a Lambda function

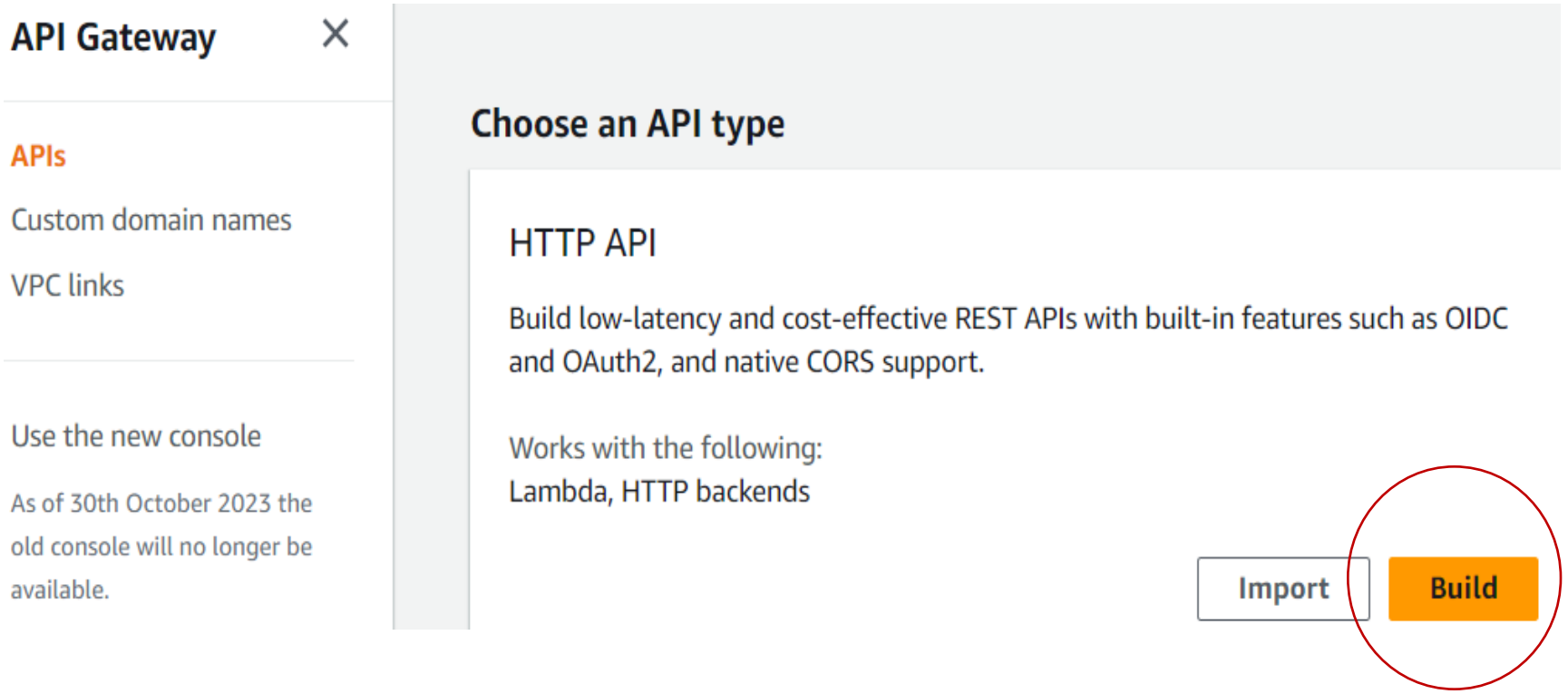


Step1: create a Lambda function



- 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



Step2.1: create and configure integrations

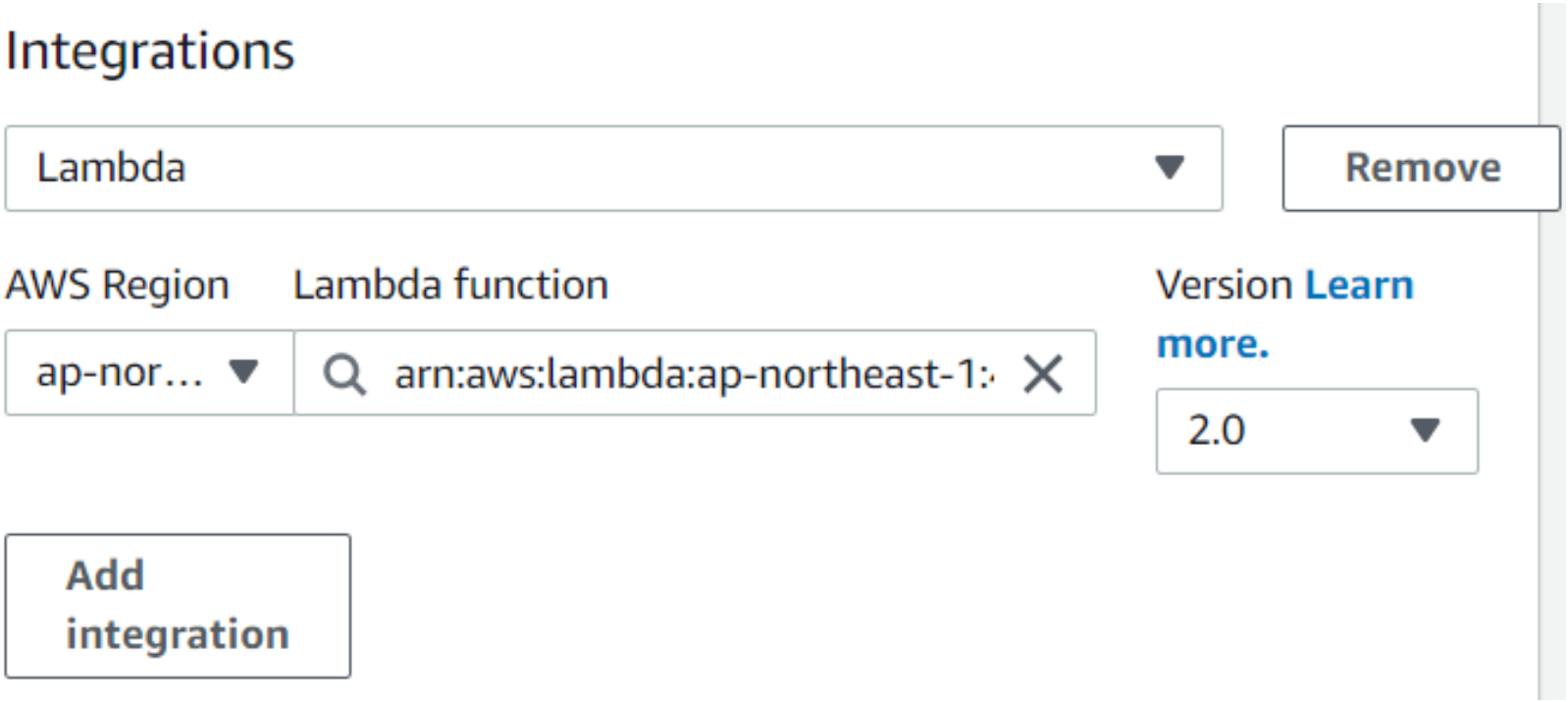
Create and configure integrations

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.

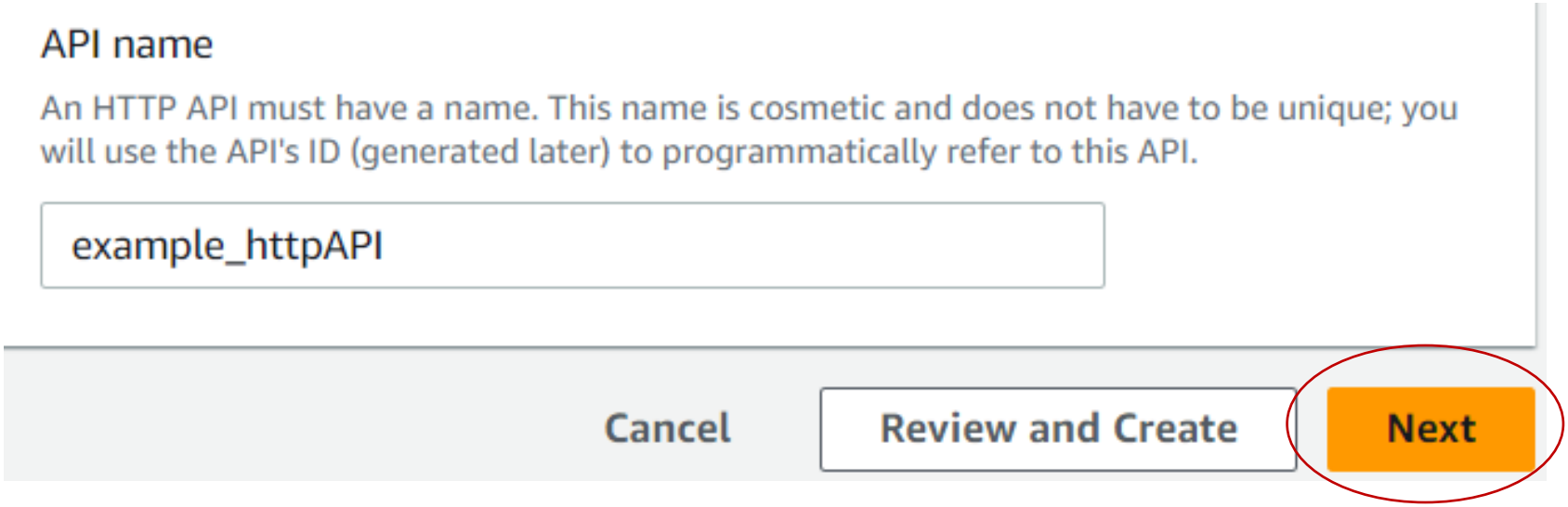
Integrations

Add integration

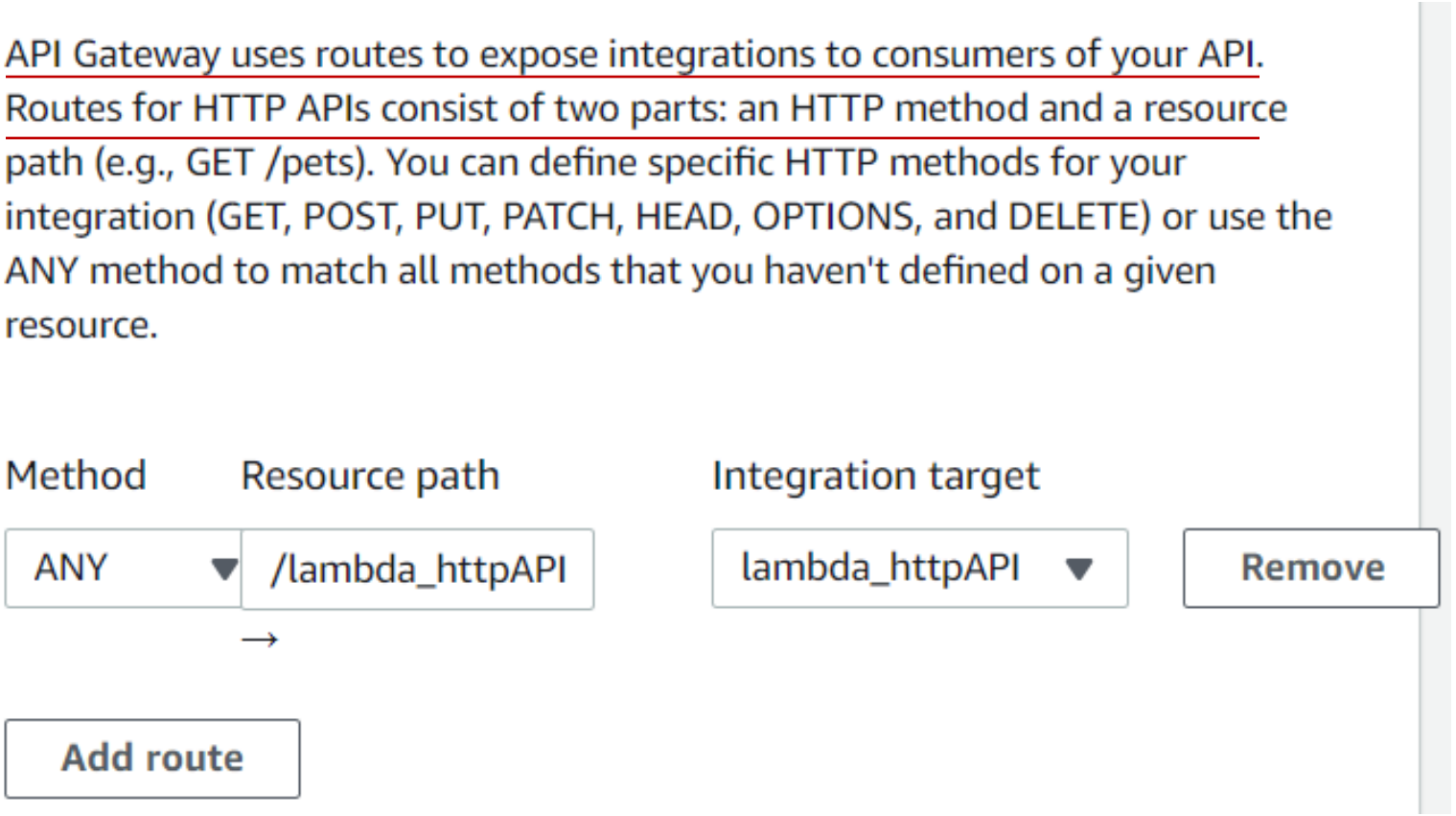
Step2.1: create and configure integrations



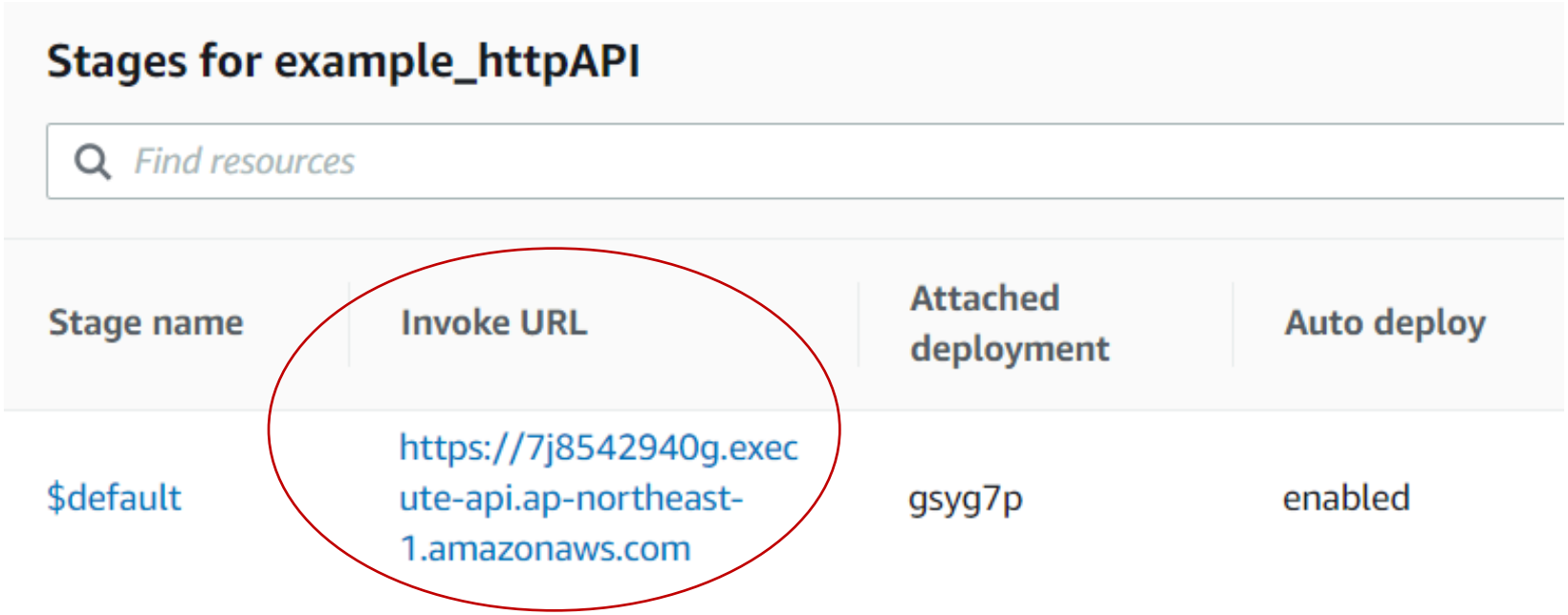
Step2.2: specify the API name



Step2.3: create routes

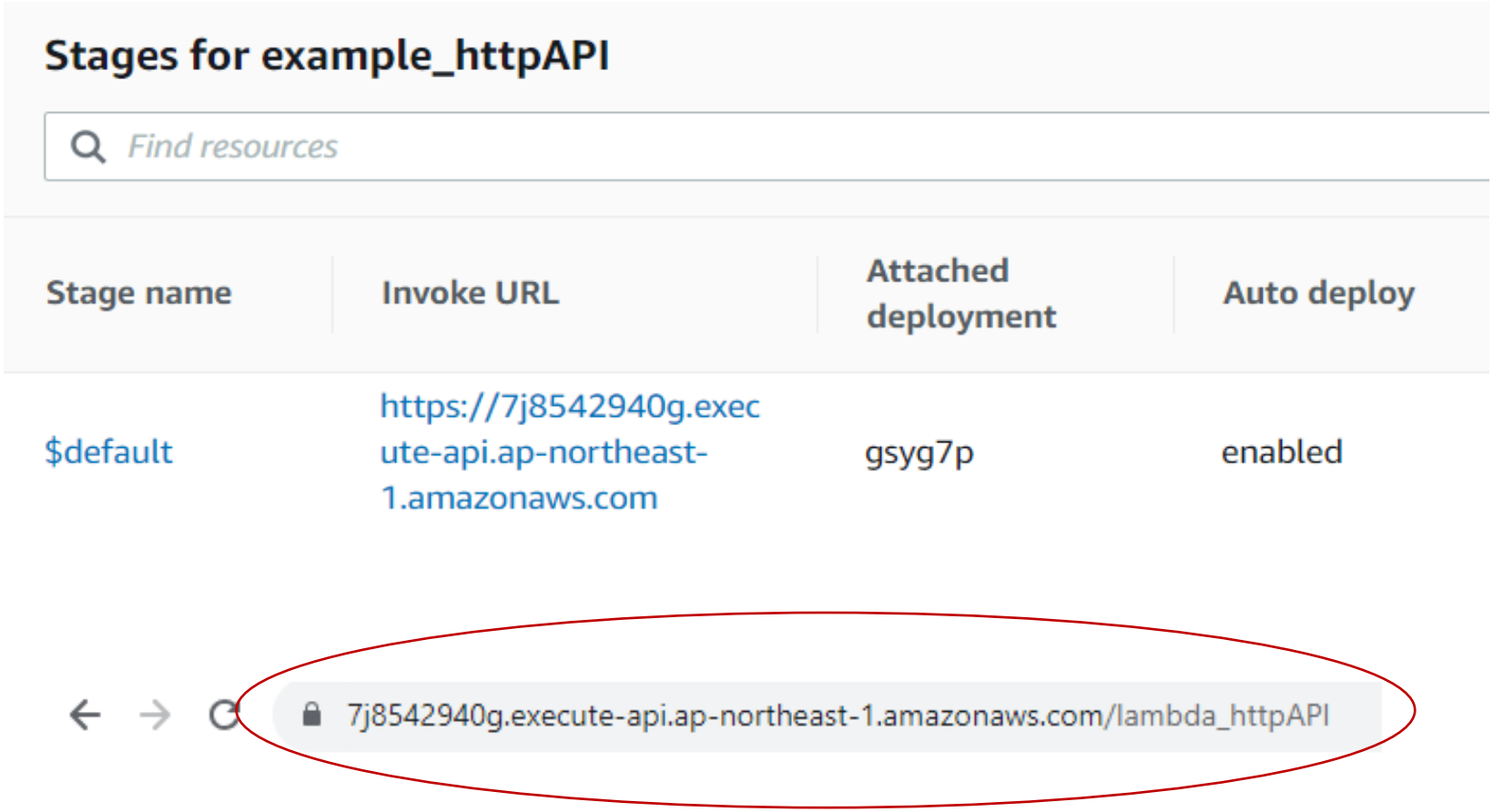


Step3: test the HTTP API



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

Step3: test the HTTP API



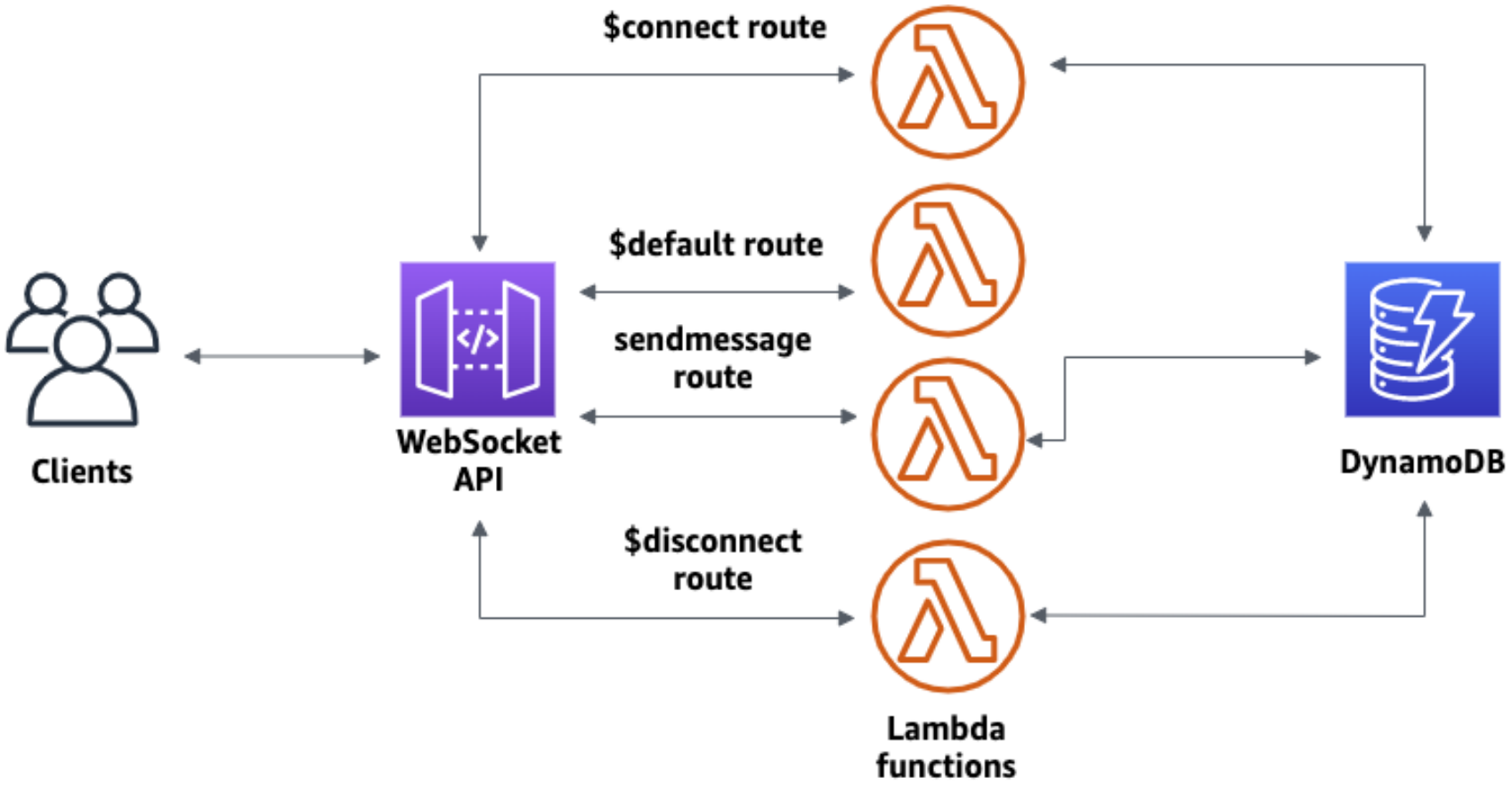
Step4: clean up

- 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

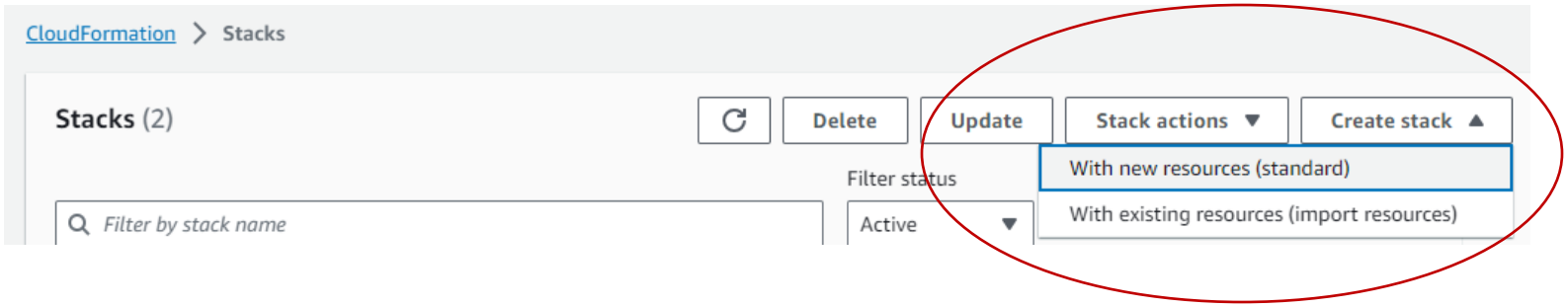


Step1: create Lambda functions and a DynamoDB table via CloudFormation

- 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.

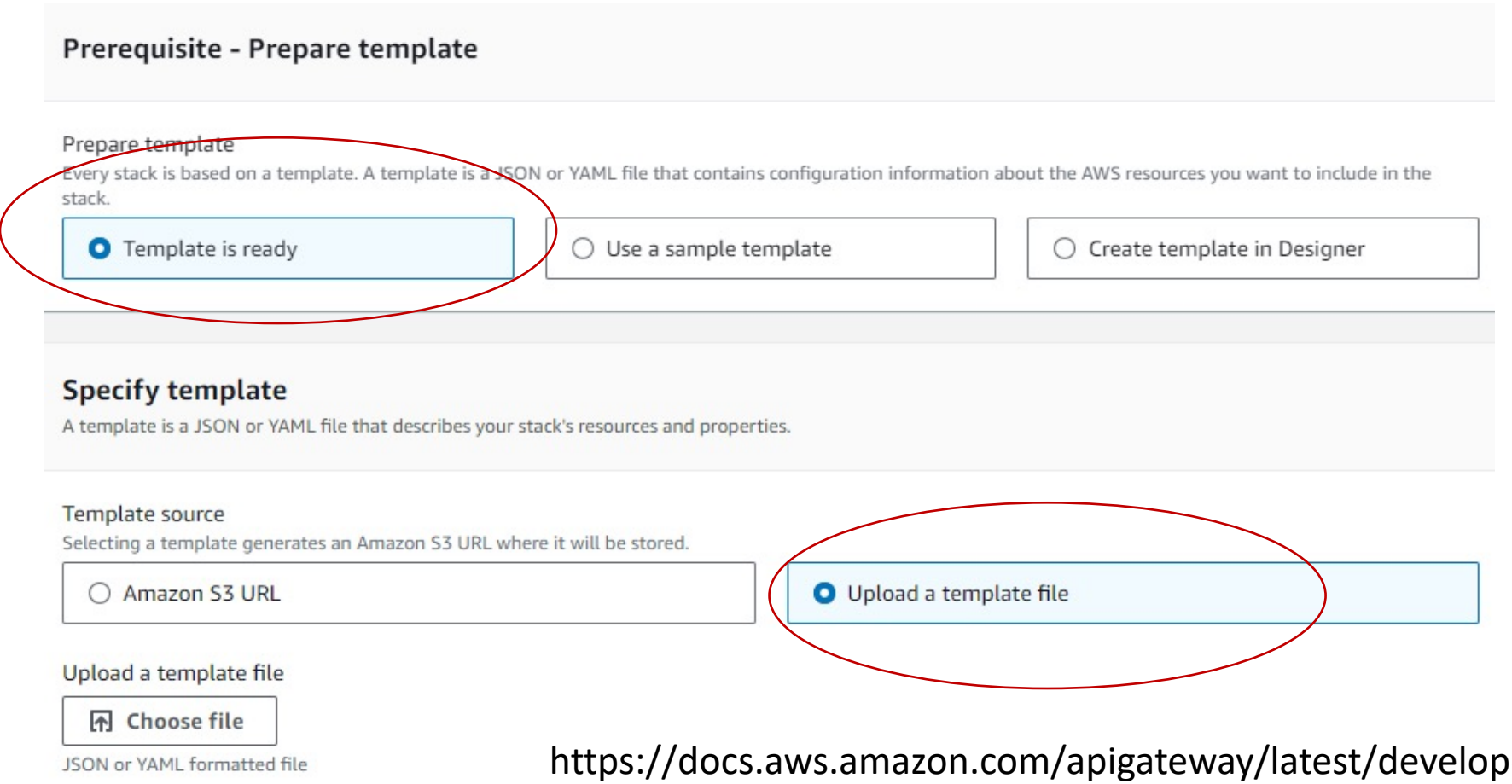
Step1.1: create Lambda functions and a DynamoDB table via CloudFormation

- 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.



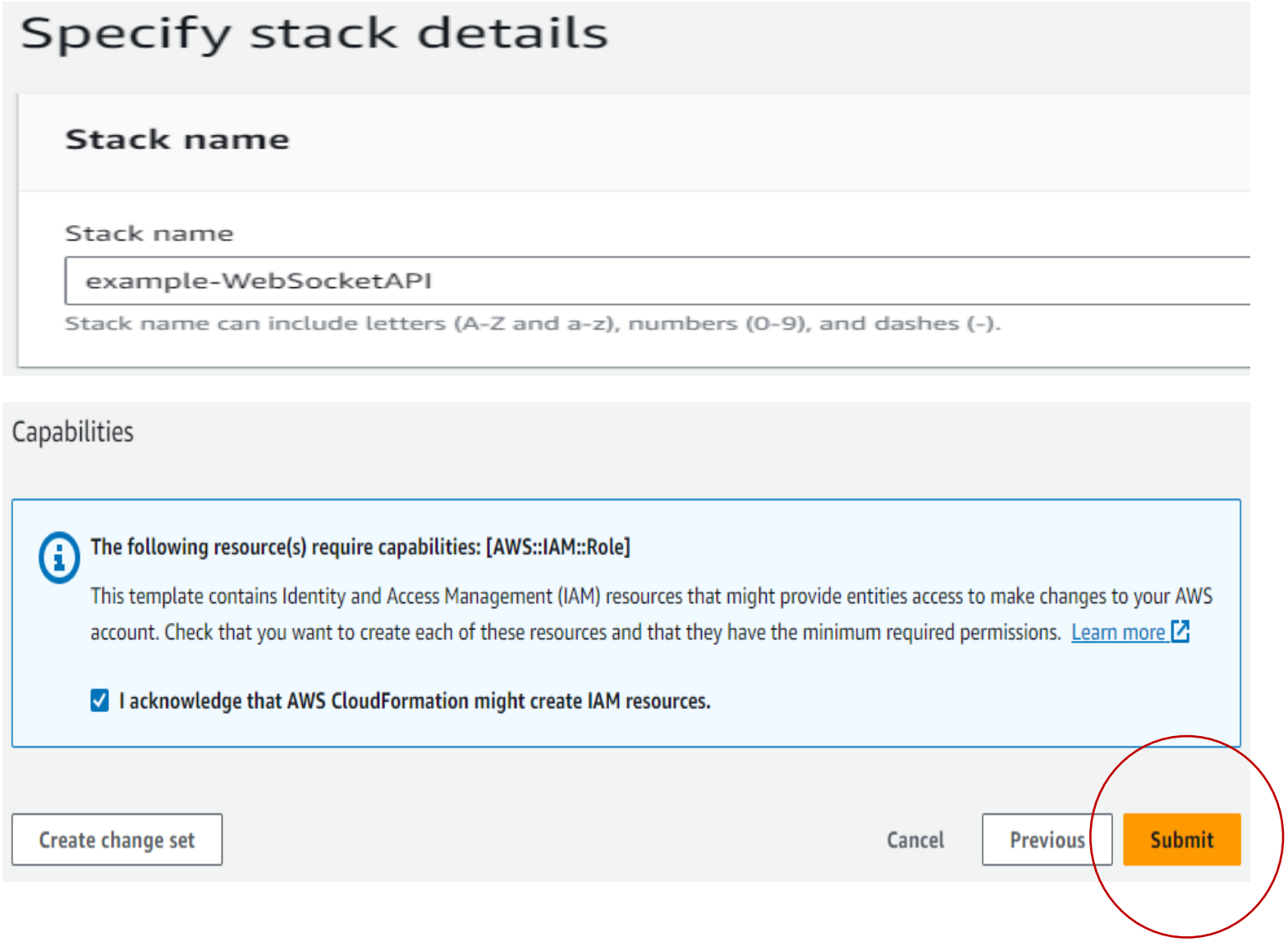
https://docs.aws.amazon.com/apigateway/latest/developerguide/samples/ws-chat-app-starter.zip

Step1.2: prepare template

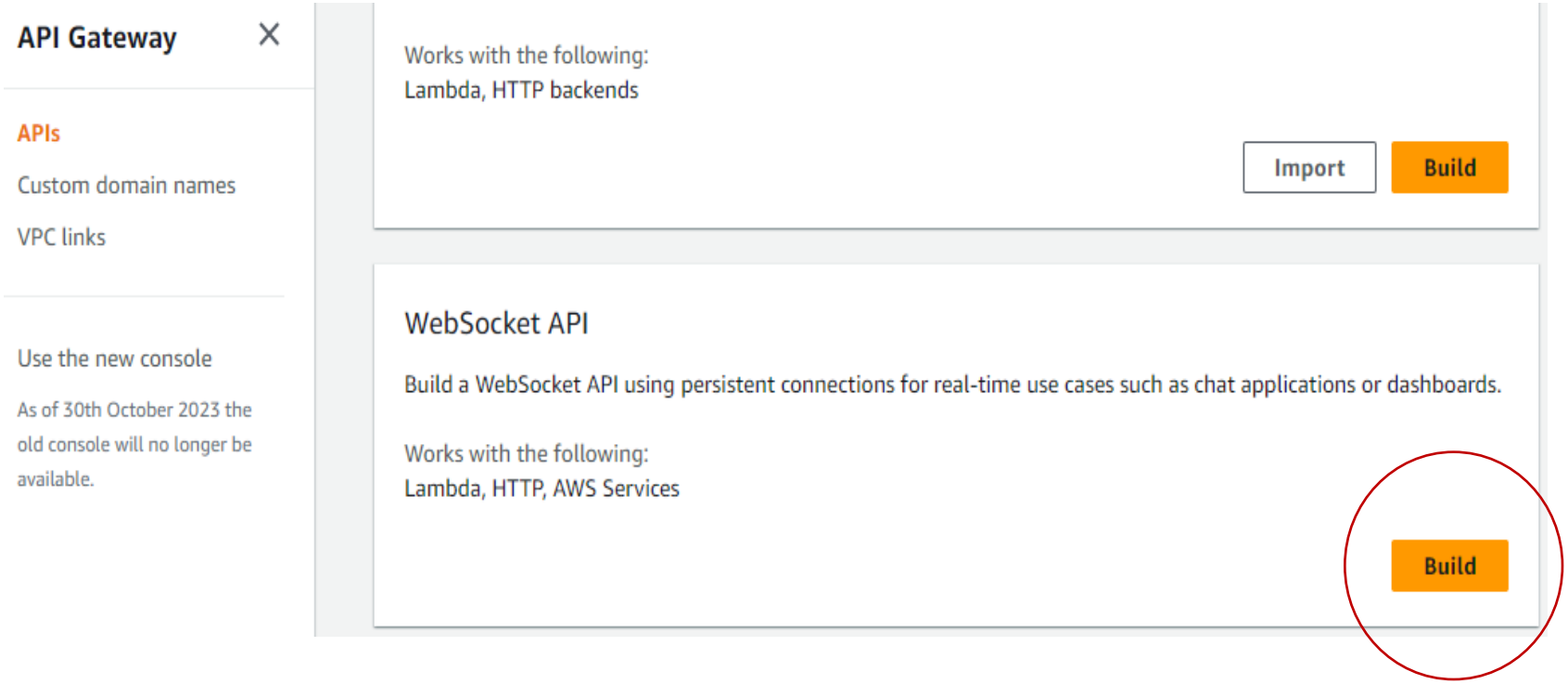
- An example template that creates a DynamoDB table and Lambda functions.

```
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
```

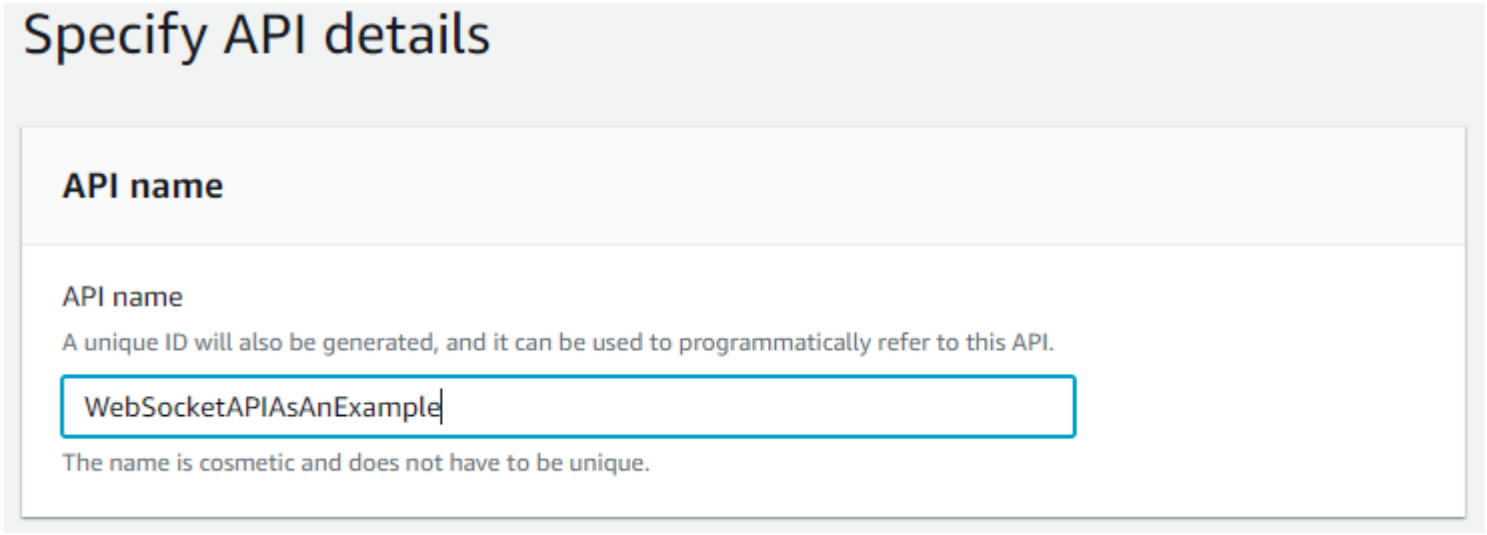
Step1.3: specify stack details



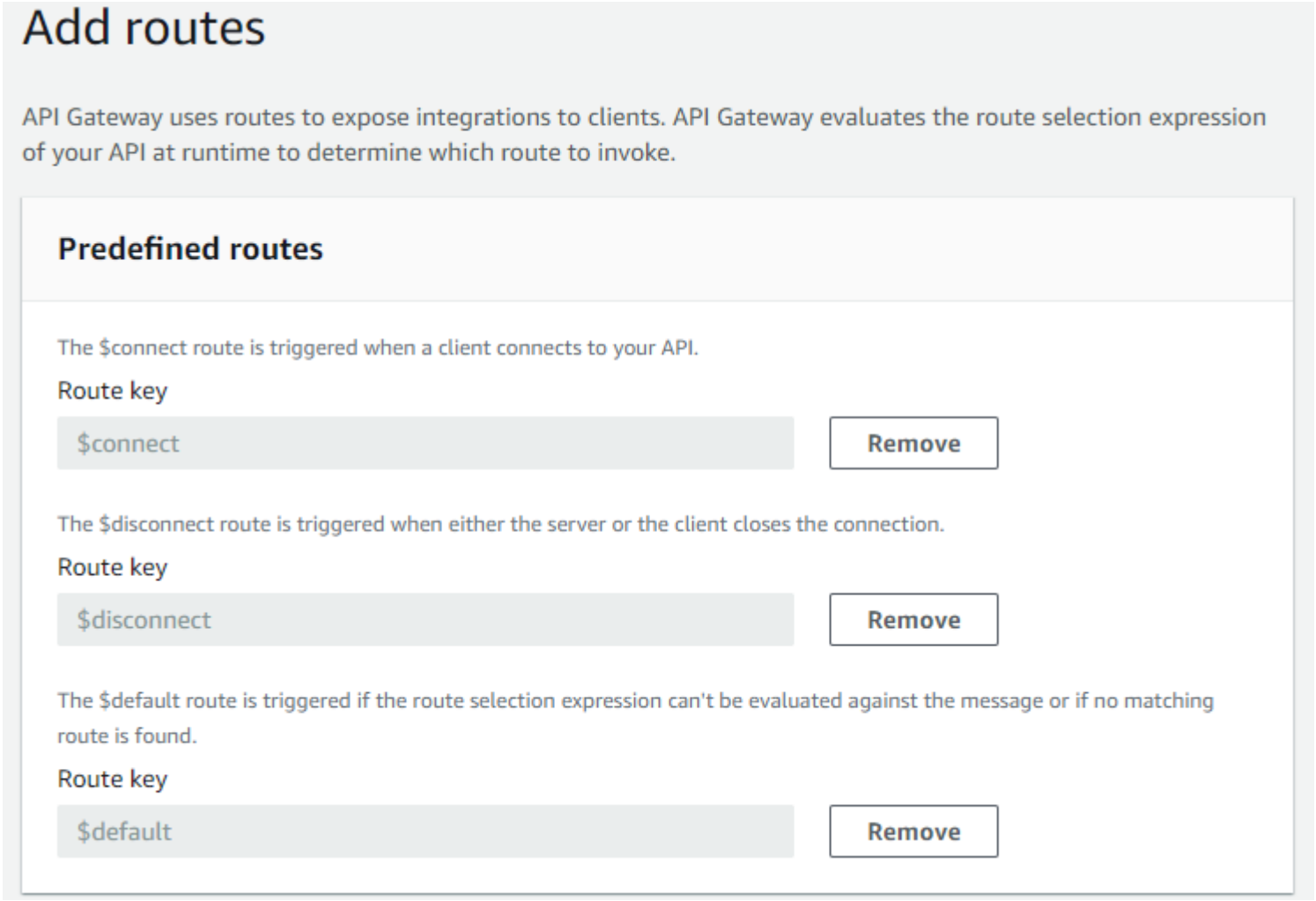
Step2: create a WebSocket API



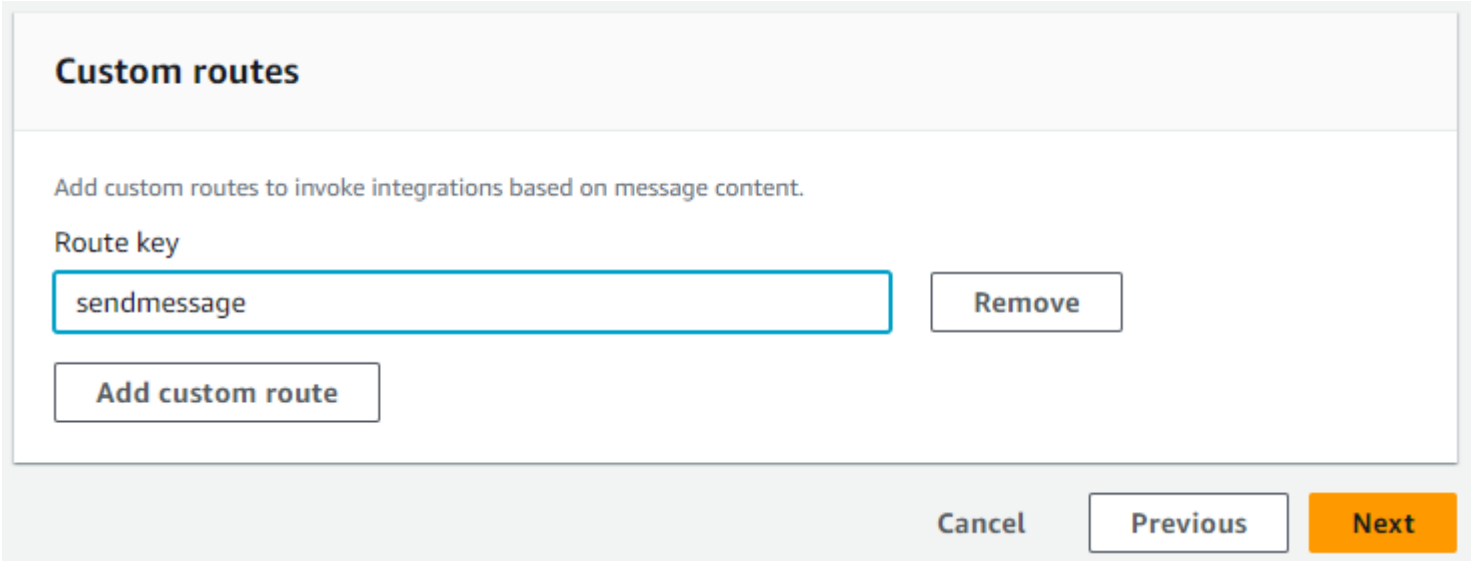
Step2.1: specify API details



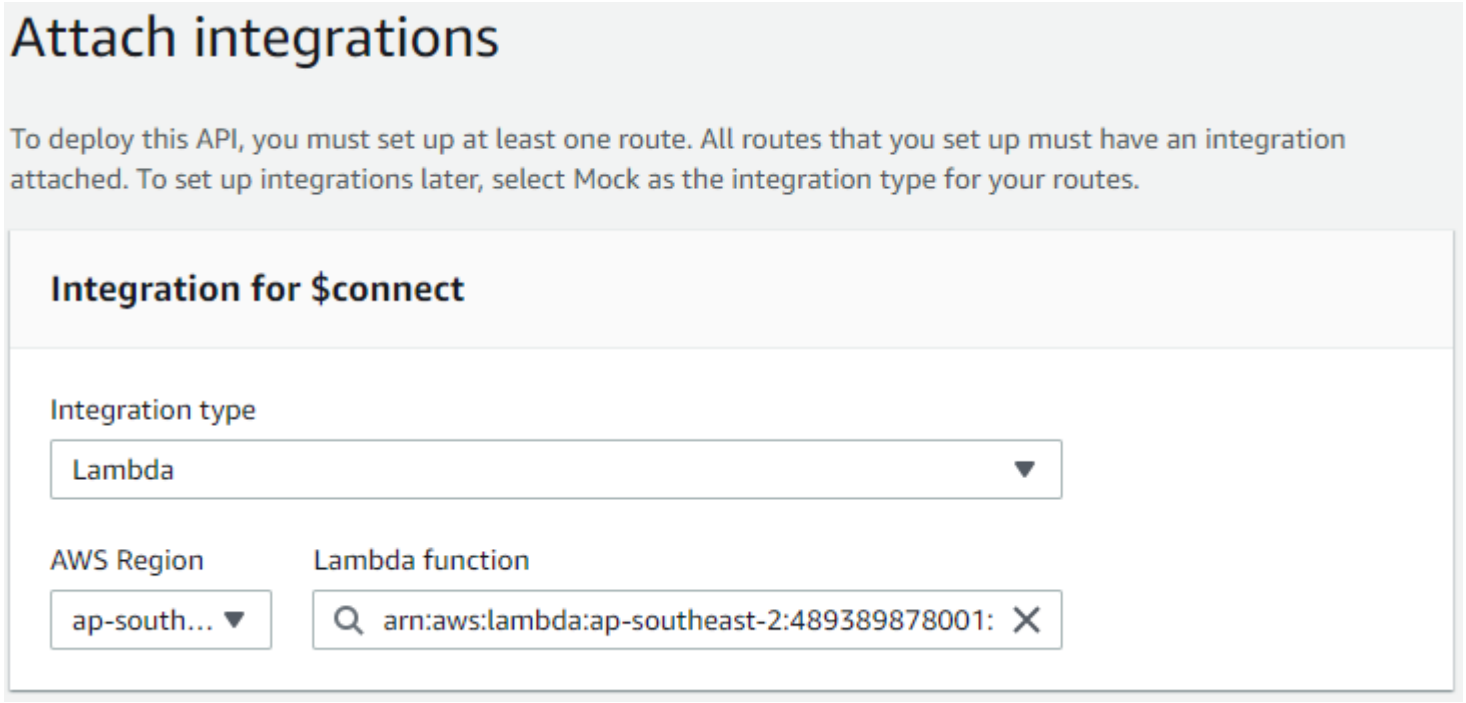
Step2.2: add predefined routes



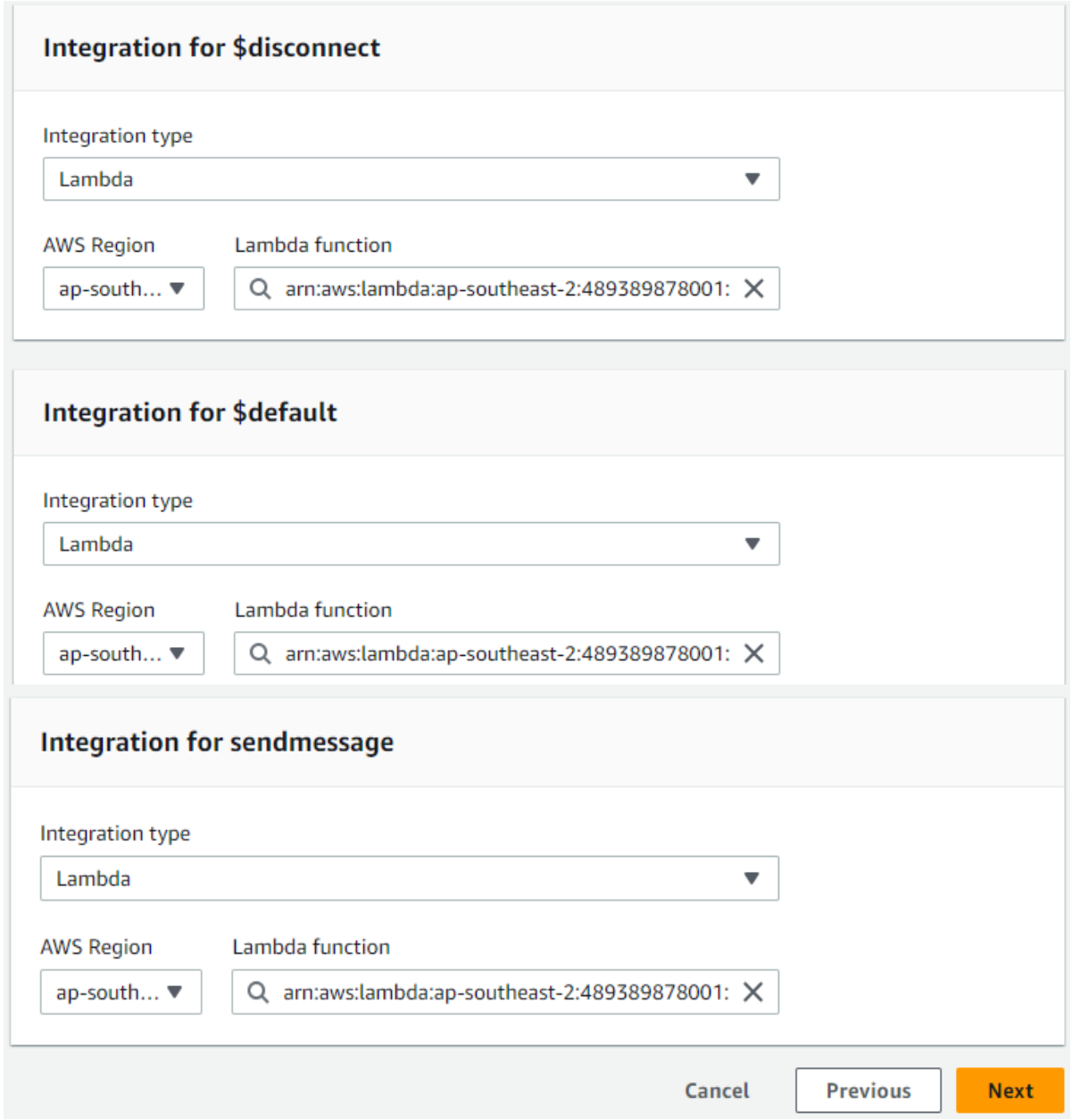
Step2.3: add custom routes



Step2.4: attach integrations



Step2.4: attach integrations



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.