

Crie sua primeira API CRUD - Progress Report Compass

Criar uma tabela no Dynamo DB

Resumo: O Amazon DynamoDB é um serviço de banco de dados NoSQL totalmente gerenciado, que oferece desempenho rápido e escalabilidade sem complicações. Ele elimina a necessidade de gerenciar tarefas administrativas como provisionamento de hardware, instalação, configuração e dimensionamento de clusters.

Passos para criar uma tabela no DynamoDB:

1. Acesse o console do DynamoDB:
<https://console.aws.amazon.com/dynamodb/>
2. Selecione "Criar tabela".
3. Nomeie a tabela como `http-crud-tutorial-items`.
4. Defina `id` como a chave primária.
5. Clique em "Criar".
6. Em alguns segundos, a tabela estará disponível.

Esta tabela pode armazenar e recuperar qualquer quantidade de dados e é usada para armazenar dados de sua API. Cada item na tabela possui um ID exclusivo, que serve como chave de partição.

Create table

Amazon DynamoDB is a fully managed non-relational with seamless scalability.

Create table

Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key is made up of one or two attributes that uniquely identify items, partition the data across each partition.

1

Table name* ⓘ

Primary key* **Partition key**

2 String ⓘ

☐ Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Auto Scaling capacity set to 70% target utilization of 5 reads and 5 writes.
- Encryption at Rest with DEFAULT encryption type

+ Add tags **NEW!**

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple. Advanced alarm settings are available in the CloudWatch management console.

Crie uma função lambda:

O AWS Lambda é um serviço que executa código sem a necessidade de gerenciar servidores. Ele escala automaticamente de acordo com a demanda, sendo ideal para criar funções que interajam com outros serviços da AWS, como o DynamoDB.

Passos para criar uma função Lambda que realiza operações CRUD no DynamoDB:

1. Acesse o console do Lambda:

<https://console.aws.amazon.com/lambda>

2. Escolha "Criar função".

3. Selecione "Autor do zero".

4. Defina o nome da função como ``http-crud-tutorial-function``.

5. Em "Informações de tempo de execução", selecione Node.js 14.x.

6. Em "Permissões", escolha "Alterar função de execução padrão".

7. Selecione "Criar uma nova função a partir de modelos de política da AWS".

8. Defina o nome da função como ``http-crud-tutorial-role``.

9. Em "Policy templates", escolha ****Simple microservice permissions****, o que permite que a função interaja com o DynamoDB.

Configuração do código da função:

1. Após criar a função, role até o editor de código-fonte no console.

2. Substitua o conteúdo do arquivo ``index.js`` pelo código fornecido, que permite as operações de ****Criar, Ler, Atualizar e Apagar**** (CRUD) itens na tabela do DynamoDB.

- O código verifica a rota recebida pelo evento do API Gateway e executa a operação correspondente no DynamoDB.

3. Após inserir o código, selecione "Implantar" para atualizar a função.

Nota: Este tutorial usa uma única função Lambda para todas as operações. Como prática recomendada, é melhor criar funções separadas para cada rota. Além disso, a política de permissões foi simplificada; recomenda-se criar uma política IAM personalizada para conceder permissões mínimas necessárias.

Basic information

Function name

Enter a name that describes the purpose of your function.

http-crud-tutorial-function

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 code editor supports only Node.js, Python, and Ruby.

Node.js 14.x

Permissions [Info](#)


By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize the role.

▼ Change default execution role

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- ☐ Create a new role with basic Lambda permissions
- ☐ Use an existing role
- ☒ Create a new role from AWS policy templates

 Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions.

Role name

Enter a name for your new role.

http-crud-tutorial-role

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

Policy templates - optional [Info](#)

Choose one or more policy templates.

Start typing ...

Simple microservice permissions X
DynamoDB

```
const AWS = require("aws-sdk");
```

```
const dynamo = new AWS.DynamoDB.DocumentClient();
```

```
exports.handler = async (event, context) => {
```

```
  let body;
```

```
  let statusCode = 200;
```

```
  const headers = {
```

```
    "Content-Type": "application/json"
```

```
  };
```

```
  try {
```

```
    switch (event.routeKey) {
```

```
      case "DELETE /items/{id}":
```

```
        await dynamo
```

```
          .delete({
```

```
            TableName: "http-crud-tutorial-items",
```

```
            Key: {
```

```
              id: event.pathParameters.id
```

```
            }
```

```
          })
```

```
          .promise();
```

```
        body = `Deleted item ${event.pathParameters.id}`;
```

```
        break;
```

```
      case "GET /items/{id}":
```

```
        body = await dynamo
```

```
          .get({
```

```
            TableName: "http-crud-tutorial-items",
```

```
            Key: {
```

```
              id: event.pathParameters.id
```

```
            }
```

```
          })
```

```

        .promise();

    break;

case "GET /items":

    body = await dynamo.scan({ TableName: "http-crud-tutorial-items" }).promise();

    break;

case "PUT /items":

    let requestJSON = JSON.parse(event.body);

    await dynamo

        .put({

            TableName: "http-crud-tutorial-items",

            Item: {

                id: requestJSON.id,

                price: requestJSON.price,

                name: requestJSON.name

            }

        })

        .promise();

    body = `Put item ${requestJSON.id}`;

    break;

default:

    throw new Error(`Unsupported route: "${event.routeKey}"`);

}

} catch (err) {

    statusCode = 400;

    body = err.message;

} finally {

    body = JSON.stringify(body);

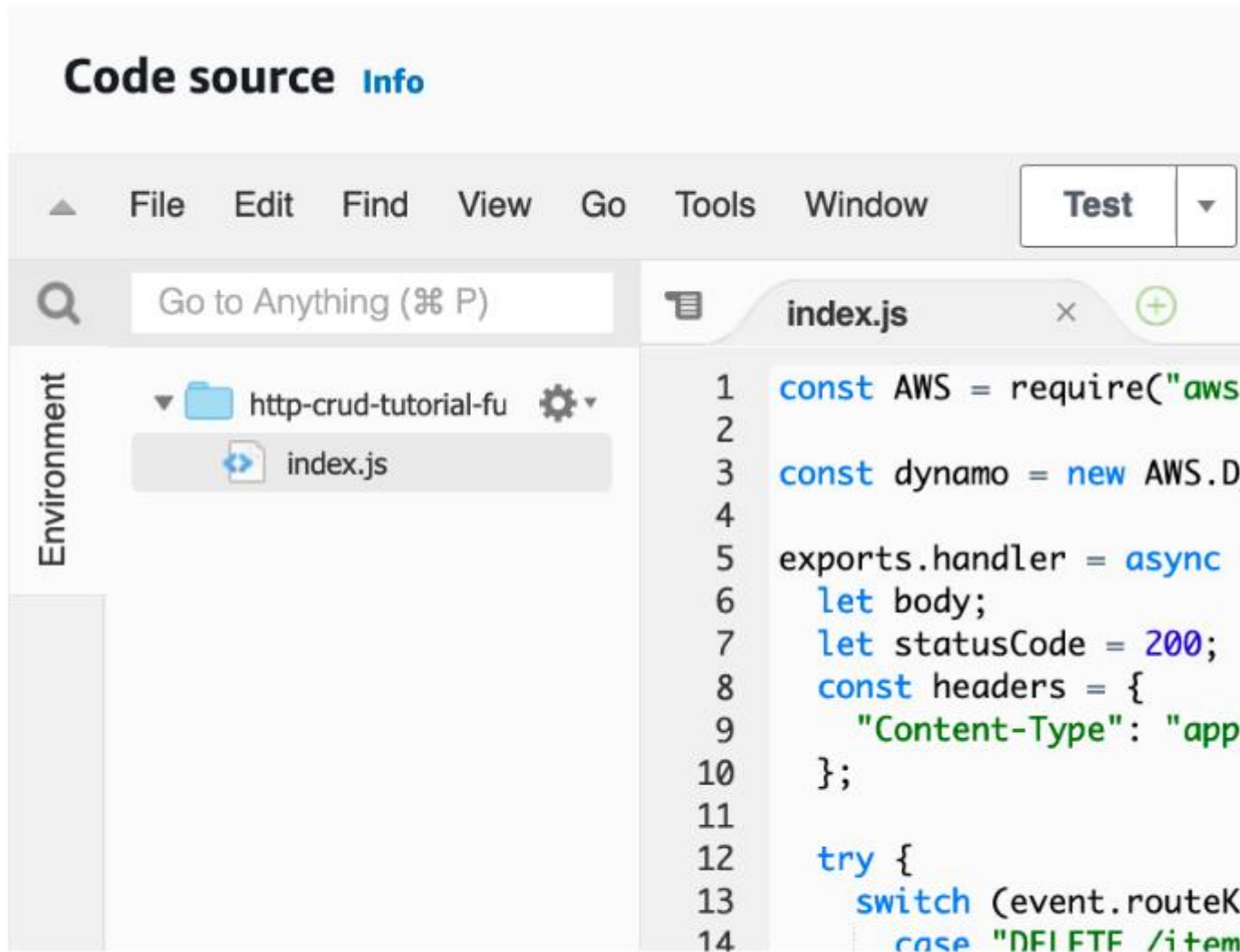
}

return {

    statusCode,

```

```
body,  
headers  
};  
};
```



Crie uma API http:

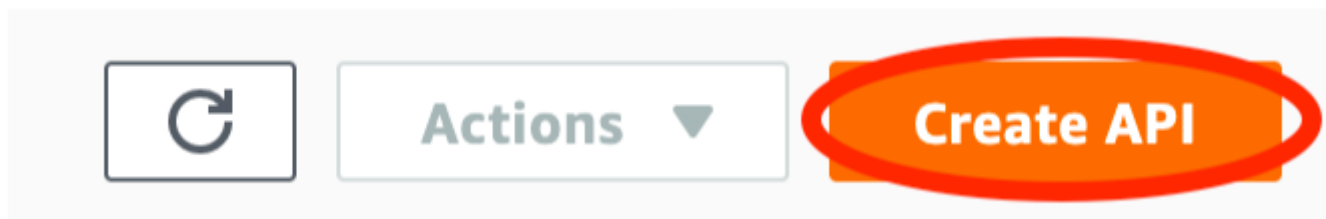
Resumo:

O Amazon API Gateway é um serviço que facilita a criação, publicação, manutenção e monitoramento de APIs em qualquer escala. Ele permite criar APIs RESTful usando APIs HTTP ou REST, e, em combinação com o AWS Lambda, forma a parte voltada para o aplicativo da infraestrutura sem servidor da AWS.

Passos para criar uma API HTTP:

1. Acesse o console do API Gateway:
https://console.aws.amazon.com/apigateway
2. Selecione "Criar API". Se você visualizar uma tela de boas-vindas, pode pular esta etapa.
3. Escolha "Build API HTTP" para criar uma API HTTP.
4. Nomeie a API como `http-crud-tutorial-api`.
5. Selecione "Avançar".
6. Na tela "Configurar rotas", escolha "Próximo" para pular a criação de rotas (isso será configurado posteriormente).
7. Revise o estágio padrão criado pelo API Gateway (`\$default`) e selecione "Avançar".
8. Selecione "Criar" para finalizar a criação da API.

Resultado: Uma API HTTP foi criada com sucesso, fornecendo um ponto de extremidade HTTP para sua função Lambda. Rotas e integrações serão configuradas em etapas posteriores para conectar a API à função Lambda.



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

API name

An HTTP API must have a name. This name is cosmetic and you will use the API's ID (generated later) to programmatically

http-crud-tutorial-api

Cancel

Review



Successfully created API http-crud-tutorial-api

Criar Rotas:

Resumo:

Rotas no API Gateway permitem que você direcione solicitações de API para recursos de backend específicos. Para a API de exemplo, você cria quatro rotas, cada uma combinando um método HTTP com um caminho de recurso.

Passos para criar rotas na API HTTP:

1. Acesse o console do API Gateway:
<https://console.aws.amazon.com/apigateway>
2. Escolha sua API, nomeada `http-crud-tutorial-api`.
3. No painel esquerdo, selecione "Rotas".
4. Clique em "Criar rota".
5. Para o Método, selecione GET e, para o caminho, insira `/items/{id}`.
 - O `{id}` é um parâmetro de caminho que será extraído da solicitação quando um cliente fizer uma chamada.
6. Clique em "Criar".

7. Repita as etapas 4 a 6 para criar as seguintes rotas adicionais:

- GET /items
- DELETE /items/{id}
- PUT /items

8. Confirme se todas as rotas foram criadas corretamente.

Resultado: Agora, sua API possui rotas configuradas para GET /items, GET /items/{id}, DELETE /items/{id}, e PUT /items, permitindo a execução das operações CRUD no backend.

API Gateway



APIs

Custom domain names

VPC links

API: http-crud-tutor...
(qf4toyicpa)

▼ Develop

Routes

Authorization

Integrations

CORS

Reimport

Export

▼ Deploy

Stages

▼ Protect

Throttling

▼ Monitor

Metrics

Logging

API Gateway > Routes

Routes

Routes for http-crud-tutorial-api

Create

🔍 Search

Route and method

Choose a method and enter a path to create a route. You can also speci

GET



/

Route and method

Choose a method and enter a path to create a route. You can also speci

DELETE



/items/{id}

Routes

Routes for http-crud-tutorial-api

Create

🔍 Search

▼ /items

GET

PUT

▼ /{id}

GET

DELETE

Criar uma Integração:

Resumo:

No API Gateway, uma integração conecta uma rota a um recurso de backend, como uma função Lambda. Para a API de exemplo, você criará uma integração Lambda que será usada para todas as rotas.

Passos para criar uma integração no API Gateway:

1. Acesse o console do API Gateway:

<https://console.aws.amazon.com/apigateway>

2. Escolha sua API, ``http-crud-tutorial-api``.
3. No painel esquerdo, selecione "Integrações".
4. Clique em "Gerenciar integrações" e, em seguida, "Criar".
5. Pule a etapa de anexar a integração a uma rota (isso será feito posteriormente).
6. Para o tipo de integração, selecione Função Lambda.
7. No campo da função Lambda, insira ``http-crud-tutorial-function``.
8. Escolha "Criar" para finalizar a integração.

Resultado: Você criou uma integração Lambda para a API, que será usada em todas as rotas. A etapa de anexar essa integração às rotas será realizada posteriormente.

▼ Develop

Routes

Authorization

Integrations

CORS

Reimport

Export

▼ Deploy

Stages

Integrati

Attach integr

Integration
crud-tutor



Integration target

Integration type

Lambda function

Integration details

Integration target

Choose the Lambda function that API Gateway invokes when the route

AWS Region

eu-west-1 ▼

Lambda function

⌕ http-crud-tutorial-func

► **Advanced settings**

Anexe sua integração às rotas:

Resumo:

Você deve anexar a integração Lambda às rotas da API para que a função Lambda seja invocada quando qualquer rota for chamada. Neste exemplo, todas as rotas da API usarão a mesma integração Lambda, formando um aplicativo monolítico.

Passos para anexar integrações às rotas no API Gateway:

1. Acesse o console do API Gateway:
https://console.aws.amazon.com/apigateway
2. Escolha sua API, `http-crud-tutorial-api`.
3. No painel esquerdo, selecione "Integrações".
4. Selecione uma rota.
5. Em "Escolher uma integração existente", selecione http-crud-tutorial-function.
6. Clique em "Anexar integração".
7. Repita as etapas 4 a 6 para todas as rotas (GET /items, GET /items/{id}, DELETE /items/{id}, PUT /items).

Resultado: Todas as rotas da API estão agora conectadas à função Lambda `http-crud-tutorial-function`. O console mostrará que a integração do AWS Lambda está anexada a cada rota. Sua API HTTP está pronta para ser testada.

▼ Develop

Routes

Authorization

Integrations

CORS

Reimport

Export

▼ Deploy

Stages

▼ Protect

Throttling

▼ Monitor

Metrics

Integrations

Attach integrations to routes

Routes for http-crud-tutorial-api

🔍 Search

▼ /items

GET

PUT

▼ /{id}

GET

DELETE

Routes for http-crud-tutorial-api

 Search

▼ /items

GET

AWS Lambda

PUT

AWS Lambda

▼ /{id}

GET

AWS Lambda

DELETE

AWS Lambda

Acesso ao Cloud9:

Resumo:

O AWS Cloud9 é um ambiente de desenvolvimento integrado (IDE) acessível via navegador, que suporta várias linguagens de programação e oferece ferramentas para codificação, construção, execução, teste e depuração de software. Com ele, você pode trabalhar localmente ou com repositórios remotos, facilitando o desenvolvimento e o lançamento de software na nuvem.

Passos para acessar e configurar o AWS Cloud9 para testar sua API:

1. Acesse o console do AWS Cloud9:
<https://console.aws.amazon.com/cloud9/>
2. Certifique-se de estar na região correta no canto superior direito.
3. Clique no botão "Criar ambiente".

4. Nomeie sua instância como `api-workshop`.
5. Configure as opções da instância conforme necessário.
6. Revise as configurações e clique em "Criar".
7. Após alguns minutos, sua instância Cloud9 será criada.

Configuração após a criação da instância:

1. Feche a tela de boas-vindas.
2. Abra um terminal em tela cheia clicando no "x" para fechar outras janelas.
3. Seu terminal no Cloud9 está agora pronto para testes de API.

Resultado: Agora, você tem um ambiente Cloud9 configurado e pronto para testar sua API, utilizando o terminal integrado.



New AWS Cloud9 environment



Create environment

Step 1

Name environment

Step 2

Configure settings

Step 3

Review

Configure settings

Environment settings

Environment type [Info](#)

Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports).

- ☒ **Create a new EC2 instance for environment (direct access)**
Launch a new instance in this region that your environment can access directly via SSH.
- ☐ **Create a new no-ingress EC2 instance for environment (access via Systems Manager)**
Launch a new instance in this region that your environment can access through Systems Manager.
- ☐ **Create and run in remote server (SSH connection)**
Configure the secure connection to the remote server for your environment.

Instance type

- ☒ **t2.micro (1 GiB RAM + 1 vCPU)**
Free-tier eligible. Ideal for educational users and exploration.
- ☐ **t3.small (2 GiB RAM + 2 vCPU)**
Recommended for small-sized web projects.
- ☐ **m5.large (8 GiB RAM + 2 vCPU)**
Recommended for production and general-purpose development.
- ☐ **Other instance type**
Select an instance type.

t3.nano

Platform

- ☒ **Amazon Linux 2 (recommended)**
- ☐ Amazon Linux 1
- ☐ Ubuntu Server 18.04 LTS

Cost-saving setting

Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After four hours

IAM role

AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

► Network settings (advanced)

No tags associated with the resource.

[Add new tag](#)

You can add 50 more tags.

[Cancel](#)[Previous step](#)[Next step](#)

AWS Cloud9

Welcome to your development environment

AWS Cloud9 allows you to write, run, and debug your code with just a browser. You can [tour the IDE](#) , write code for [AWS Lambda](#) and [Amazon API Gateway](#) , share your IDE with others in real time, and much more.

Toolkit for AWS Cloud9

The AWS Toolkit for Cloud9 is an IDE extension that simplifies accessing and interacting with resources from services such as AWS Lambda, AWS CloudFormation, and AWS API Gateway. With the toolkit, developers can also develop, debug, and deploy applications using the AWS Serverless Application Model (SAM). [Learn more](#)

Support

If you have any questions or experience issues, refer to our documentation or reach us to get help.

[Documentation](#)

[Get Help](#)

[Security Best Practices](#)

Getting started

[Create File](#)

[Upload File](#)

[Clone from](#)

Configure AWS Cloud9

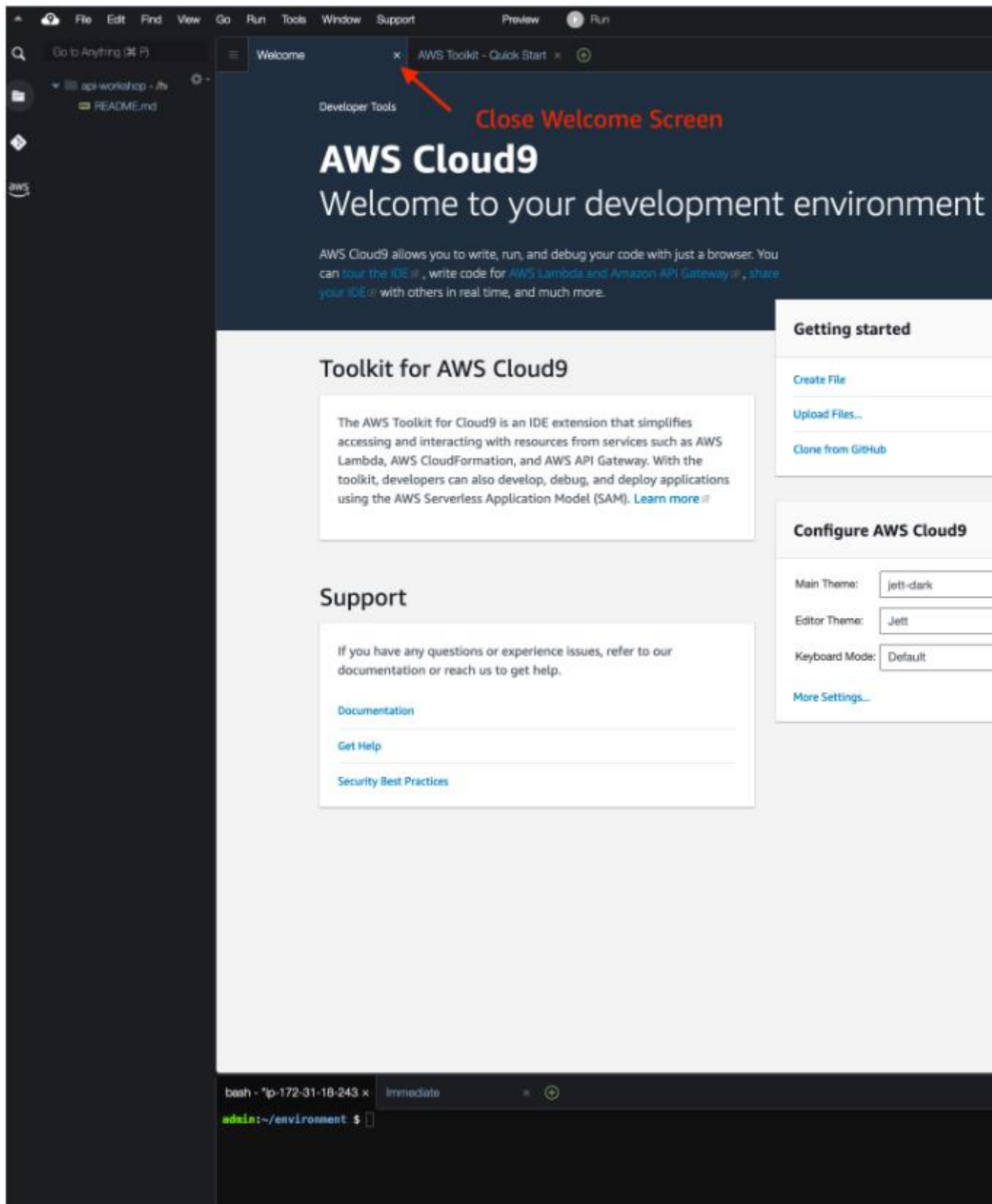
Main Theme:

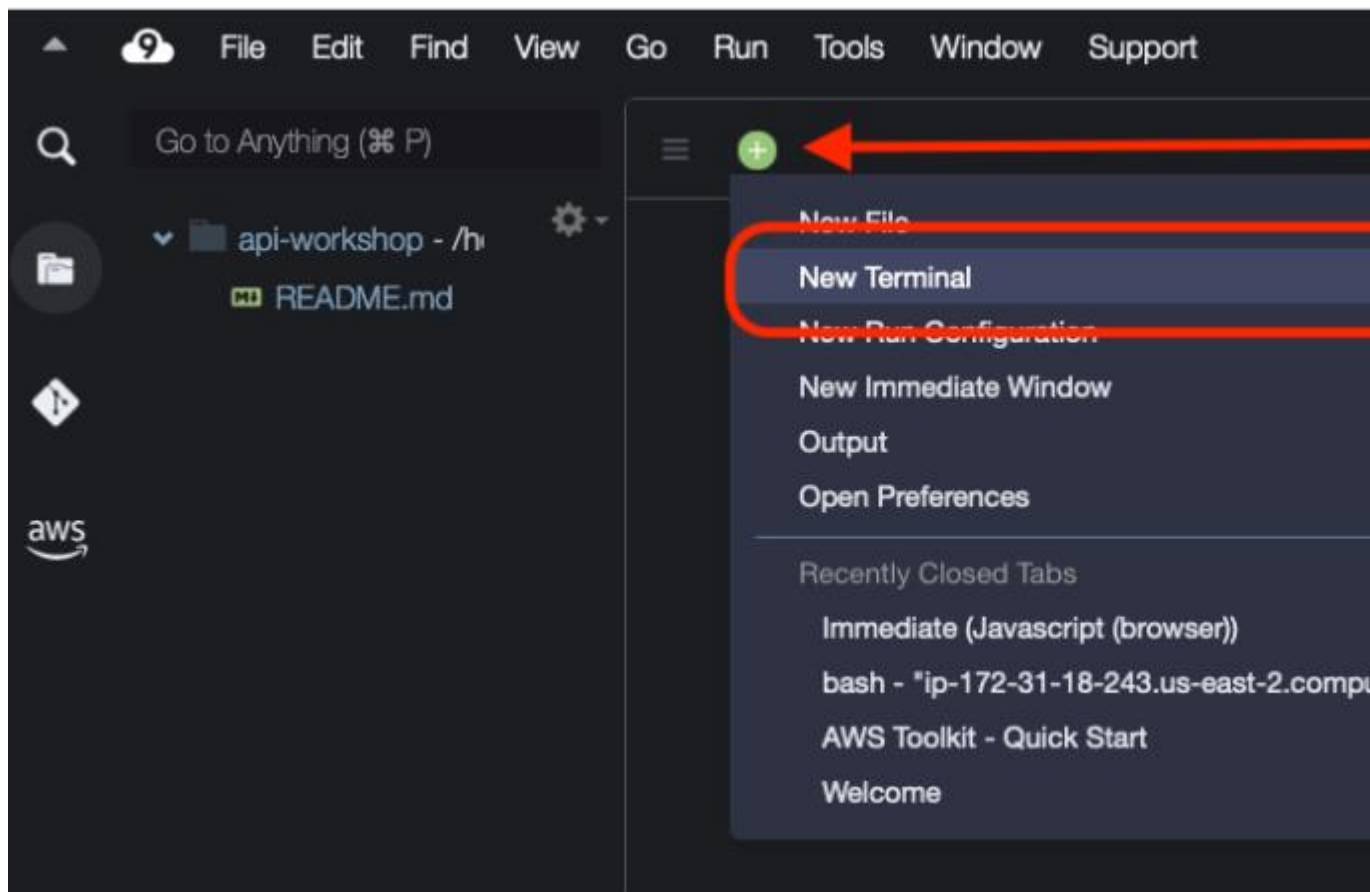
Editor Theme:

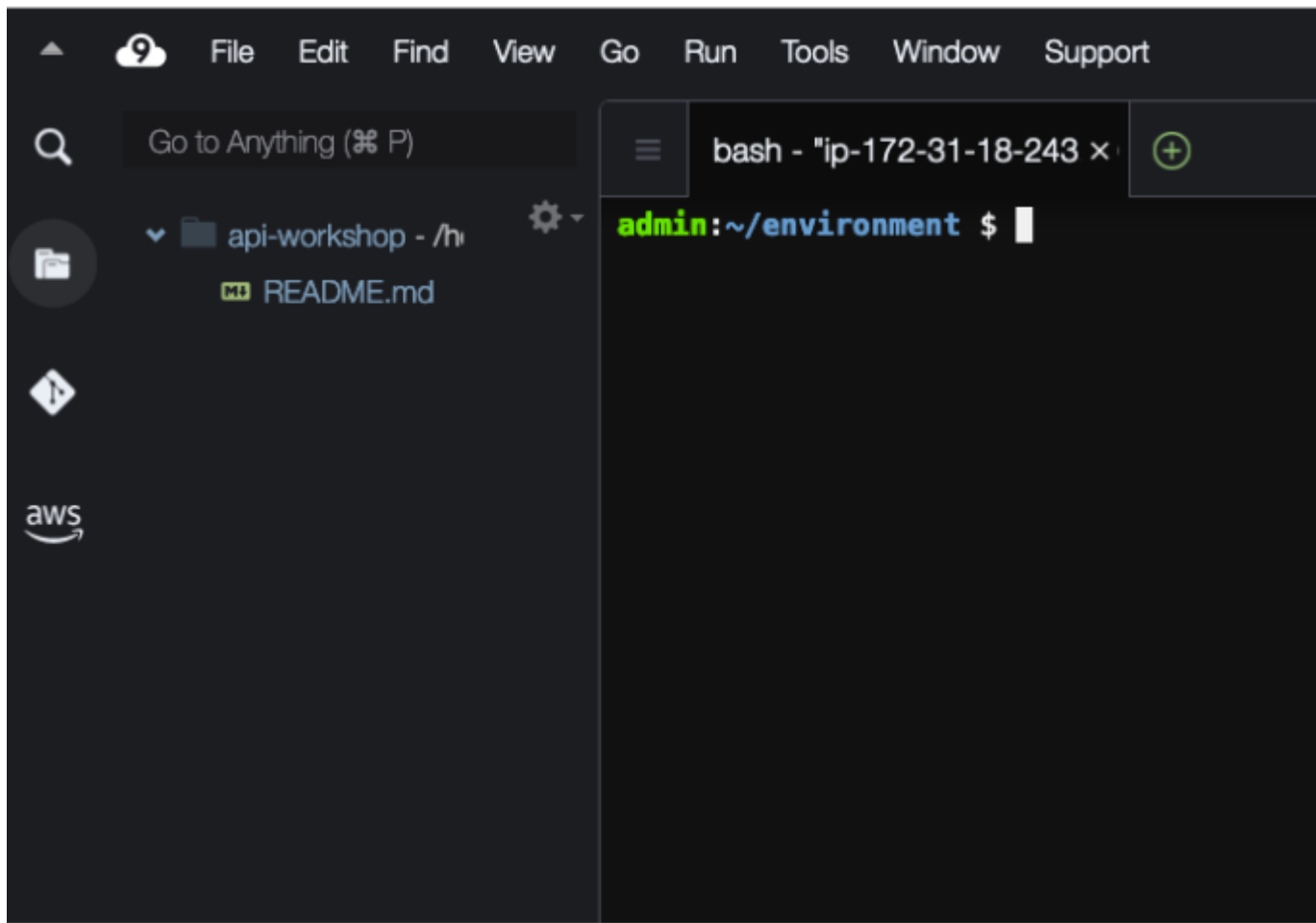
Keyboard Mode:

[More Settings...](#)

We are creating your AWS Cloud9 environment.
This can take a few minutes.







Teste sua API:

Resumo Completo com Códigos:

Para garantir que sua API está funcionando corretamente, você pode usar a ferramenta de linha de comando `curl` para testar as operações CRUD. Siga os passos abaixo para realizar esses testes.

1. Obter o URL de Invocação da API

1. Acesse o console do API Gateway:

<https://console.aws.amazon.com/apigateway/>

2. Escolha sua API.

3. Copie o URL de invocação da sua API, que aparece em ****Invoke URL**** na página ****Details****.

- Exemplo de URL de invocação: `https://abcdef123.execute-api.eu-west-1.amazonaws.com`

2. Conecte-se ao AWS Cloud9

1. Navegue até

<https://console.aws.amazon.com/cloud9/>.

2. Abra o IDE do Cloud9.

3. No terminal do Cloud9, defina uma variável com o URL de invocação:

```
```bash
```

Substitua pelo seu URL de invocação real

```
export INVOKE_URL="https://abcdef123.execute-api.eu-west-1.amazonaws.com"
```

```
```
```

3. Criar ou Atualizar um Item

Use o comando abaixo para criar ou atualizar um item na sua tabela DynamoDB. O comando inclui um corpo de solicitação com o ID, preço e nome do item:

```
```bash
```

```
curl -X "PUT" -H "Content-Type: application/json" -d '{
```

```
 "id": "abcdef234",
```

```
 "price": 12345,
```

```
 "name": "myitem"
```

```
}' $INVOKE_URL/items
```

```
```
```

4. Obter Todos os Itens

Use o comando abaixo para listar todos os itens na tabela:

```
```bash
```

```
curl -s $INVOKE_URL/items | js-beautify
```

```
```
```

5. Obter um Item Específico

Para obter um item específico pelo seu ID, use:

```
```bash
```

```
curl -s $INVOKE_URL/items/abcdef234 | js-beautify
```

```
```
```

6. Excluir um Item

Use o comando a seguir para excluir um item pelo ID:

```
```bash
```

```
curl -X "DELETE" $INVOKE_URL/items/abcdef234
```

```
```
```

7. Verificar se o Item foi Excluído

Finalmente, obtenha todos os itens novamente para verificar se o item foi removido:

```
```bash
curl -s $INVOKE_URL/items | js-beautify
```
```

Resultado: Após executar esses comandos, você terá confirmado que sua API está funcionando corretamente, permitindo criar, ler, atualizar e excluir itens na tabela DynamoDB.

[API Gateway](#) > [Details](#)

http-crud-tutorial-api

API details

| | |
|----------------|------------------|
| API ID | Protocol |
| abcdef123 | HTTP |
| Description | Default endpoint |
| No Description | Enabled |

Stages for http-crud-tutorial-api

| Stage name | Invoke URL |
|------------|---|
| \$default | https://abcdef123.execute-api.eu-west-1.amazonaws.com |

Limpar:

Resumo Completo com Códigos e Instruções para Limpeza dos Recursos:

Após completar os testes da sua API e para evitar custos desnecessários, siga os passos abaixo para excluir os recursos criados durante o exercício. Isso inclui a tabela do DynamoDB, a API HTTP, a função Lambda e outros recursos associados.

1. Excluir uma Tabela do DynamoDB

1. Acesse o console do DynamoDB:

<https://console.aws.amazon.com/dynamodb/>

2. Selecione sua tabela.

3. Clique em Excluir tabela.

4. Confirme sua escolha e clique em Excluir.

2. Excluir uma API HTTP

1. Entre no console do API Gateway:

<https://console.aws.amazon.com/apigateway/>

2. Na página APIs, selecione sua API.

3. Clique em Actions e depois em Delete.

4. Selecione Excluir para confirmar.

3. Excluir uma Função Lambda

1. Acesse o console do Lambda:

<https://console.aws.amazon.com/lambda/>

2. Na página Functions, selecione sua função.

3. Clique em Actions e depois em Delete.

4. Confirme a exclusão clicando em Excluir.

4. Excluir um Grupo de Log da Função Lambda

1. Acesse o console do Amazon CloudWatch:

<https://console.aws.amazon.com/cloudwatch/>

2. Abra a página Grupos de log.

3. Selecione o grupo de log da função, por exemplo, `/aws/lambda/http-crud-tutorial-function`.

4. Clique em Actions e depois em **Delete log group.

5. Confirme a exclusão.

5. Excluir a Função de Execução de uma Função Lambda

1. Acesse o console do AWS Identity and Access Management:

<https://console.aws.amazon.com/iam/>

2. Na página Funções, selecione a função associada, por exemplo, `http-crud-tutorial-role`.

3. Clique em Excluir função.

4. Selecione Sim, excluir para confirmar.

6. Excluir o Cloud9 IDE

1. Entre no console do Cloud9:

<https://console.aws.amazon.com/cloud9/>

2. Na página Your environments, selecione o IDE que você criou.

3. Escolha Delete.

4. Para confirmar sua escolha, digite `Delete`.

5. Selecione Excluir.

Resultado:

Após seguir esses passos, todos os recursos criados durante o exercício serão excluídos, ajudando a evitar custos desnecessários.