Author: Madhurima Rawat

Deploying Cloud Server Using LocalStack and EC2

This document explains how to deploy a cloud server using LocalStack, AWS API Gateway, and EC2, with a detailed breakdown of commands and outputs.

1. Creating a Resource in API Gateway

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway create-resource \
    --rest-api-id rbx2kdpyxl \
    --parent-id regpu0mm3f \
    --path-part "flaskapp"
```

Explanation:

- `aws apigateway create-resource` → Creates a new resource in an API Gateway.
- `--endpoint-url=http://localhost:4566` → Specifies the LocalStack endpoint, as we are using a local AWS environment.
- `--rest-api-id rbx2kdpyxl` → The API Gateway identifier where this resource will be added.
- `--parent-id regpu0mm3f` → Specifies the parent resource under which the new resource will be created.
- `--path-part "flaskapp"` → Defines the URL path for the new resource (i.e., `/flaskapp`).

Output:

```
| CreateResource |
+-------------+--------------+------------+-----------+
| id          | parentId     | path       | pathPart  |
+-------------+--------------+------------+-----------+
| bsw1umubix  | regpu0mm3f   | /flaskapp  | flaskapp  |
```

Breakdown of Output:

- `id` → Unique identifier (`bsw1umubix`) assigned to the newly created resource.
- `parentId` → Shows the parent resource ID (`regpu0mm3f`).
- `path` → Displays the full path of the new resource (`/flaskapp`).

- `pathPart` → Represents the last segment of the path (`flaskapp`).

2. Creating an HTTP Method (GET) for the Resource

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway put-method \
    --rest-api-id rbx2kdpyxl \
    --resource-id bsw1umubix \
    --http-method GET \
    --authorization-type NONE
```

Explanation:

- `aws apigateway put-method` → Adds an HTTP method (`GET`) to a resource in API Gateway.
- `--endpoint-url=http://localhost:4566` → Directs the command to the LocalStack environment.
- `--rest-api-id rbx2kdpyxl` → Specifies the API Gateway in which to define the method.
- `--resource-id bsw1umubix` → Identifies the resource (`/flaskapp`) to which this method applies.
- `--http-method GET` → Defines the HTTP method (`GET` in this case).
- `--authorization-type NONE` → Indicates that no authentication is required for this API method.

Output:

```
-----------------------------------------------------
|                     PutMethod                     |
+----------------+---------------------+-------------+
| apiKeyRequired |  authorizationType  | httpMethod  |
+----------------+---------------------+-------------+
|  False         |  NONE               |  GET        |
+----------------+---------------------+-------------+
```

Breakdown of Output:

- `apiKeyRequired` → Indicates whether an API key is needed (`False` means no API key is required).
- `authorizationType` → Shows the authentication type (`NONE` , meaning public access is allowed).
- `httpMethod` → Displays the HTTP method assigned to the resource (`GET`).

3. Integrating API Gateway with a Backend Service

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway put-integration \
    --rest-api-id rbx2kdpyxl \
    --resource-id bsw1umubix \
    --http-method GET \
    --integration-http-method GET \
    --type HTTP_PROXY \
    --uri http://localhost:5000/
```

Explanation:

- `aws apigateway put-integration` → Configures the integration of API Gateway with an external HTTP backend.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack instead of AWS.
- `--rest-api-id rbx2kdpyxl` → Identifies the API Gateway.
- `--resource-id bsw1umubix` → Links the integration to the `/flaskapp` resource.
- `--http-method GET` → Specifies that this integration applies to `GET` requests.
- `--integration-http-method GET` → Defines the method used to invoke the backend.
- `--type HTTP_PROXY` → Specifies a direct proxy integration.
- `--uri http://localhost:5000/` → Defines the backend endpoint to which requests should be forwarded.

Output:

```
-----------------------------------------------------
|                  PutIntegration                   |
+----------------------+----------------------------+
|  cacheNamespace      |  bsw1umubix                |
|  connectionType      |  INTERNET                  |
|  httpMethod          |  GET                       |
|  passthroughBehavior |  WHEN_NO_MATCH             |
|  timeoutInMillis     |  29000                     |
|  type                |  HTTP_PROXY                |
|  uri                 |  http://localhost:5000/    |
+----------------------+----------------------------+
```

4. Deploying the API Gateway

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway create-deployment \
    --rest-api-id rbx2kdpyxl \
    --stage-name prod
```

Explanation:

- `aws apigateway create-deployment` → Deploys the configured API Gateway.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack instead of AWS.
- `--rest-api-id rbx2kdpyxl` → Identifies the API Gateway to deploy.
- `--stage-name prod` → Creates a deployment under the `prod` stage.

Output:

```
--------------------------------
|       CreateDeployment       |
+---------------+--------------+
|  createdDate  |     id       |
+---------------+--------------+
|  1738825514.0 |  fpmrktu41t  |
+---------------+--------------+
```

Summary

These steps successfully set up an API Gateway resource `/flaskapp`, linked it to a `GET` method, integrated it with a backend service, and deployed it under the `prod` stage using LocalStack.

5. Retrieving API Gateway Stage Information

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway get-stage \
    --rest-api-id rbx2kdpyxl \
    --stage-name prod
```

Explanation:

- `aws apigateway get-stage` → Retrieves information about a specific deployment stage in API Gateway.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack instead of AWS.
- `--rest-api-id rbx2kdpyxl` → Specifies the API Gateway for which the stage details are needed.

- `--stage-name prod` → Retrieves details for the `prod` stage.

Output:

```
-----------------------------------------
|               GetStage                |
+----------------------+----------------+
|  cacheClusterEnabled |  False         |
|  cacheClusterStatus  |  NOT_AVAILABLE |
|  deploymentId        |  fpmrktu41t    |
|  description         |                |
|  stageName           |  prod          |
|  tracingEnabled      |  False         |
+----------------------+----------------+
```

Breakdown of Output:

- `cacheClusterEnabled` → Indicates if caching is enabled (`False` means caching is not used).
- `cacheClusterStatus` → Shows the status of the cache cluster (`NOT_AVAILABLE` as caching is disabled).
- `deploymentId` → ID of the deployed stage (`fpmrktu41t`).
- `description` → Empty field, as no description was provided.
- `stageName` → Name of the deployed stage (`prod`).
- `tracingEnabled` → Indicates whether AWS X-Ray tracing is enabled (`False` means disabled).

6. Testing the Flask Application

Command:

```
curl http://localhost:5000/instance-stats
```

Explanation:

- `curl` → Sends a request to the specified URL.
- `http://localhost:5000/instance-stats` → Calls the Flask application endpoint to fetch instance statistics.

Output:

{"Instance ID":"i-6c9d5e3fa4c23d261","Instance Type":"t2.micro","Public IP":"54.214.36.25","Re

Breakdown of Output:

- `Instance ID` → Unique identifier of the running EC2 instance.
- `Instance Type` → Specifies the EC2 instance type (`t2.micro`).
- `Public IP` → Shows the public IP address assigned to the instance.
- `Region` → Displays the AWS region (`us-east-1`).
- `State` → Current state of the instance (`running`).

7. Retrieving API Gateway Resources

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway get-resources \
    --rest-api-id rbx2kdpyxl
```

Explanation:

- `aws apigateway get-resources` → Fetches all resources associated with the specified API Gateway.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack instead of AWS.
- `--rest-api-id rbx2kdpyxl` → Identifies the API Gateway.

Output:

```
-------------------------------------------------------------
|                       GetResources                        |
+-----------------------------------------------------------+
||                          items                          ||
|+-------------------------+------------------------------+|
||  id                     |  regpu0mm3f                  ||
||  parentId               |                              ||
||  path                   |  /                           ||
||  pathPart               |                              ||
|+-------------------------+------------------------------+|
||                          items                          ||
|+-------------------------+------------------------------+|
||  id                     |  bsw1umubix                  ||
||  parentId               |  regpu0mm3f                  ||
||  path                   |  /flaskapp                   ||
```

```
||  pathPart              |  flaskapp                    ||
|+------------------------+------------------------------+|
|||                   resourceMethods                    |||
||+----------------------------------------------------+||
||||                         GET                        ||||
|||+-----------------------------------+---------------+|||
||||  apiKeyRequired                   |  False        ||||
||||  authorizationType                |  NONE         ||||
||||  httpMethod                       |  GET          ||||
|||+-----------------------------------+---------------+|||
|||||                methodIntegration                 |||||
||||+----------------------+--------------------------+|||||
|||||  cacheNamespace      |  bsw1umubix              |||||
|||||  connectionType      |  INTERNET                |||||
|||||  httpMethod          |  GET                     |||||
|||||  passthroughBehavior |  WHEN_NO_MATCH           |||||
|||||  timeoutInMillis     |  29000                   |||||
|||||  type                |  HTTP_PROXY              |||||
|||||  uri                 |  http://localhost:5000/  |||||
||||+----------------------+--------------------------+|||||
```

Breakdown of Output:

Resources:

- The first resource (`id: regpu0mm3f`):
  - Represents the root (`/`) of the API Gateway.
- The second resource (`id: bsw1umubix`):
  - Represents the `/flaskapp` endpoint.

Methods (GET):

- `apiKeyRequired` → No API key required (`False`).
- `authorizationType` → Publicly accessible (`NONE`).
- `httpMethod` → Specifies that `GET` is available.

Integration Details:

- `cacheNamespace` → Identifies the cache scope.
- `connectionType` → The API Gateway is connected to an external HTTP service (`INTERNET`).
- `httpMethod` → The backend request method is `GET` .
- `passthroughBehavior` → Determines how the request is handled if it doesn't match a known route (`WHEN_NO_MATCH`).
- `timeoutInMillis` → API request timeout is set to `29,000 ms` (29 seconds).
- `type` → Uses `HTTP_PROXY` , meaning requests are forwarded to the backend without modifications.

- `uri` → The backend service URL (`http://localhost:5000/`), which points to the Flask app.

8. Creating a Resource for Instance Stats

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway create-resource \
    --rest-api-id rbx2kdpyxl \
    --parent-id bsw1umubix \
    --path-part "instance-stats"
```

Explanation:

- `aws apigateway create-resource` → Creates a new resource within an API Gateway.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack instead of AWS.
- `--rest-api-id rbx2kdpyxl` → Specifies the API Gateway to add the resource.
- `--parent-id bsw1umubix` → Identifies `/flaskapp` as the parent resource.
- `--path-part "instance-stats"` → Creates a subpath under `/flaskapp` .

Output:

```
-----------------------------------------------------------------------------
|                             CreateResource                                |
+------------+-------------+---------------------------+------------------+
|     id     |  parentId   |           path            |    pathPart      |
+------------+-------------+---------------------------+------------------+
|  glsyl3hnnj|  bsw1umubix |  /flaskapp/instance-stats  |  instance-stats  |
+------------+-------------+---------------------------+------------------+
```

Breakdown of Output:

- `id` → Unique identifier for the new resource (`glsyl3hnnj`).
- `parentId` → Shows the parent resource ID (`bsw1umubix`).
- `path` → Displays the full path (`/flaskapp/instance-stats`).
- `pathPart` → Represents the last segment of the path (`instance-stats`).

9. Adding a GET Method to the Instance Stats Resource

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway put-method \
    --rest-api-id rbx2kdpyxl \
    --resource-id glsyl3hnnj \
    --http-method GET \
    --authorization-type NONE
```

Explanation:

- `aws apigateway put-method` → Defines an HTTP method for a resource.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack.
- `--rest-api-id rbx2kdpyxl` → Identifies the API Gateway.
- `--resource-id glsyl3hnnj` → Specifies the `/flaskapp/instance-stats` resource.
- `--http-method GET` → Adds a `GET` method.
- `--authorization-type NONE` → No authentication is required.

Output:

```
-------------------------------------------------------
|                     PutMethod                       |
+----------------+---------------------+--------------+
| apiKeyRequired |  authorizationType  | httpMethod   |
+----------------+---------------------+--------------+
|  False         |  NONE               |  GET         |
+----------------+---------------------+--------------+
```

Breakdown of Output:

- `apiKeyRequired` → No API key required (`False`).
- `authorizationType` → Publicly accessible (`NONE`).
- `httpMethod` → Specifies that `GET` is available.

10. Integrating API Gateway with the Instance Stats Backend

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway put-integration \
    --rest-api-id rbx2kdpyxl \
    --resource-id glsyl3hnnj \
    --http-method GET \
    --integration-http-method GET \
```

```
    --type HTTP_PROXY \
    --uri http://localhost:5000/instance-stats
```

Explanation:

- `aws apigateway put-integration` → Configures API Gateway to forward requests to a backend service.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack.
- `--rest-api-id rbx2kdpyxl` → Identifies the API Gateway.
- `--resource-id glsyl3hnnj` → Links integration to `/flaskapp/instance-stats`.
- `--http-method GET` → Applies the integration to `GET` requests.
- `--integration-http-method GET` → Uses `GET` for backend requests.
- `--type HTTP_PROXY` → Passes requests directly to the backend.
- `--uri http://localhost:5000/instance-stats` → Specifies the Flask backend service URL.

Output:

```
-------------------------------------------------------------
|                        PutIntegration                     |
+----------------------+------------------------------------+
|  cacheNamespace      |  glsyl3hnnj                        |
|  connectionType      |  INTERNET                          |
|  httpMethod          |  GET                               |
|  passthroughBehavior |  WHEN_NO_MATCH                     |
|  timeoutInMillis     |  29000                             |
|  type                |  HTTP_PROXY                        |
|  uri                 |  http://localhost:5000/instance-stats  |
+----------------------+------------------------------------+
```

Breakdown of Output:

- `cacheNamespace` → Identifies the cache for this resource.
- `connectionType` → The API Gateway is connected to an external HTTP service (`INTERNET`).
- `httpMethod` → The backend request method is `GET`.
- `passthroughBehavior` → Defines how unmatched requests are handled (`WHEN_NO_MATCH`).
- `timeoutInMillis` → API request timeout is `29,000 ms`.
- `type` → Uses `HTTP_PROXY`, meaning requests are forwarded without modifications.
- `uri` → The backend service URL (`http://localhost:5000/instance-stats`).

11. Deploying the Updated API Gateway

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway create-deployment \
    --rest-api-id rbx2kdpyxl \
    --stage-name prod
```

Explanation:

- `aws apigateway create-deployment` → Deploys the latest changes to API Gateway.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack.
- `--rest-api-id rbx2kdpyxl` → Identifies the API Gateway.
- `--stage-name prod` → Deploys under the `prod` stage.

Output:

```
---------------------------------
|       CreateDeployment        |
+---------------+---------------+
|  createdDate  |     id        |
+---------------+---------------+
|  1738826210.0 |  8bku2hvst8   |
+---------------+---------------+
```

Breakdown of Output:

- `createdDate` → Timestamp of the deployment (`1738826210.0`).
- `id` → Unique identifier for this deployment (`8bku2hvst8`).

12. Verifying the Integration of Instance Stats API

Command:

```
aws --endpoint-url=http://localhost:4566 apigateway get-integration \
    --rest-api-id rbx2kdpyxl \
    --resource-id glsyl3hnnj \
    --http-method GET
```

Explanation:

- `aws apigateway get-integration` → Retrieves integration details for a specific API method.
- `--endpoint-url=http://localhost:4566` → Uses LocalStack.

- `--rest-api-id rbx2kdpyxl` → Identifies the API Gateway.
- `--resource-id glsyl3hnnj` → Specifies the `/flaskapp/instance-stats` resource.
- `--http-method GET` → Retrieves integration details for `GET`.

Output:

```
-------------------------------------------------------------
|                      GetIntegration                       |
+----------------------+------------------------------------+
|  cacheNamespace      |  glsyl3hnnj                        |
|  connectionType      |  INTERNET                          |
|  httpMethod          |  GET                               |
|  passthroughBehavior |  WHEN_NO_MATCH                     |
|  timeoutInMillis     |  29000                             |
|  type                |  HTTP_PROXY                        |
|  uri                 |  http://localhost:5000/instance-stats  |
+----------------------+------------------------------------+
```

Breakdown of Output:

- Confirms that API Gateway successfully integrates with the Flask backend at `/instance-stats`.
- Shows that all `GET` requests will be forwarded to `http://localhost:5000/instance-stats`.

These steps successfully set up an API Gateway endpoint `/flaskapp/instance-stats`, linked it to a `GET` method, integrated it with the Flask backend, and deployed it under the `prod` stage.