cle**Create DynamoDB Table**

1. Navigate to DynamoDB in the AWS Management Console.
2. Create a table called friends:
   1. Set the partition key to id of type String.
   2. Check **Add sort key**.
   3. Set the sort key value to name of type String.
3. Leave **Use default settings** checked.
4. Click **Create**.
5. Wait until the **Table status** reads **Active**.
6. Log into the EC2 instance with SSH, using the credentials provided.
7. Clone the repository: git clone https://github.com/linuxacademy/content-aws-sam.
8. Change to the directory for this lab: cd content-aws-sam/labs/Creating-Serverless-Application-S3-HTTP-API-Lambda-DynamoDB.
9. Seed the DynamoDB table with data: ./dynamodb.py friends.

check\_circle**Create Lambda Function**

1. Navigate to Lambda in the AWS Management Console.
2. Click **Create function**.
3. Leave **Author from scratch** selected.
4. Set *Function name* to **GetFriends**.
5. Set *Runtime* to **Python 3.8**.
6. Under *Permissions*, expand **Choose or create an execution role**.
7. Select **Create a new role from AWS policy templates**.
8. Set *Role name* to *GetFriendsRole*.
9. Under *Policy templates* select **Simple microservice permissions**.
10. Click **Create function**.
11. In the *Function code* editor, replace the boilerplate code with the contents of <https://github.com/linuxacademy/content-aws-sam/blob/master/labs/Creating-Serverless-Application-S3-HTTP-API-Lambda-DynamoDB/lambda_function.py>.
12. Under *Environment variables* click **Manage environment variables**.
13. Click **Add environment variable**.
14. Set Key to TABLE\_NAME.
15. Set Value to friends.
16. Click **Save**.
17. Click **Save** (Lambda function).

check\_circle**Create HTTP API**

1. Navigate to API Gateway in the AWS Management Console.
2. Click **Build** under the *HTTP API* API type.
3. Click **Add integration**.
4. Under *Integration type* select **Lambda**.
5. Under *Integration target* select **us-east-1**.
6. Choose the Lambda function named **GetFriends**.
7. Under *API name*, enter **GetFriends**.
8. Click **Review and Create**.
9. Click **Create**.
10. In the left navigation menu, under *Develop*, click **CORS**.
11. Click **Edit**.
12. Under *Access-Control-Allow-Origin* enter **\***.
13. Click **Add**.
14. Click **Save**.
15. Click **Deploy**.
16. Under *Select a stage* enter **$default**.

check\_circle**Create S3 Bucket**

1. Navigate to S3 in the AWS Management Console.
2. Select **Create bucket**.
3. Under *Bucket name* enter a ***globally unique*** name.
4. Under *Region* select **US East (N. Virginia) us-east-1**.
5. Uncheck **Block *all* public access**.
6. Check the acknowledgement box.
7. Click **Create bucket**.
8. Click the bucket name.
9. Under *Properties* click **Static website hosting**.
10. Select **Use this bucket to host a website**.
11. Under *index document* enter **index.html**.
12. Click **Save**.
13. Under **Bucket Policy**, paste the contents of <https://github.com/linuxacademy/content-aws-sam/blob/master/labs/Creating-Serverless-Application-S3-HTTP-API-Lambda-DynamoDB/bucket_policy.json>.
14. Replace **my-bucket-name** with the name of your bucket.
15. Click **Save**.

check\_circle**Configure and Deploy Web Application**

1. On the EC2 instance, change to the app directory: cd app.
2. Edit app.js and replace the value of **invokeUrl** with the HTTP API Invoke URL.
3. Save the file.
4. Copy the application to S3: aws s3 sync . s3://<bucket name>.
5. Browse to the S3 website URL *http://.s3-website-us-east-1.amazonaws.com*