Let’s have a look at the step by step approach to copy the JSON file from AWS S3 to Redshift:

* [Step 1: Create and Upload JSON File to S3](https://hevodata.com/learn/json-to-redshift/" \l "c1)
* [Step 2: Create JSONPath File](https://hevodata.com/learn/json-to-redshift/" \l "c2)
* [Step 3: Load the Data into Redshift](https://hevodata.com/learn/json-to-redshift/" \l "c3)

#### Step 1: Create and Upload JSON File to S3

To get started with the Redshift JSON extraction process, let’s see how you will create and upload the JSON files to S3.

* **Step 1**: Create a sample JSON file named employee.json and add a few sample records of the employee as below.

{“emp\_id” : 1, “emp\_name” : “Sam”, “emp\_dept” : “HR”}

{“emp\_id” : 2, “emp\_name” : “Jib”, “emp\_dept” : “Tech”}

{“emp\_id” : 3, “emp\_name” : “Nick”, “emp\_dept” : “IT”}

{“emp\_id” : 4, “emp\_name” : “Joseph”, “emp\_dept” : “Tech”}

* **Step 2**: Upload the file to AWS S3 using AWS CLI.

Syntax - aws s3 cp <source file> <target S3 path>

Example - aws s3 cp employee.json s3://test-bucket/json/

#### Step 2: Create JSONPath File

JSONPath is an expression that specifies the path to a single element in a JSON hierarchical data structure. Amazon Redshift doesn’t support any JSONPath features, such as wildcard characters or filter expressions, and only expects the complete JSONPath of that element to parse it correctly. As a result, Amazon Redshift can’t parse complex, multi-level data structures.

1. Create the JSONPath file for the above JSON. Copy the below content and save it in a file as – employee\_jsonpath.json.

{

"jsonpaths": [

"$.emp\_id",

"$emp\_name",

"$emp\_dept"

]

}

1. Use AWS CLI to upload the file into S3.

Syntax - aws s3 cp <source file> <target S3 path>

Example - aws s3 cp employee\_jsonpath.json s3://test-bucket/jsonpaths/

#### Step 3: Load the Data into Redshift

Let’s see what are the steps to load data into Redshift.

1. First, you have to create table DDL in the Redshift, which will hold the schema information of the JSON.

CREATE TABLE employee(emp\_id INT, emp\_name STRING, emp\_dept STRING, primary key(emp\_id))

1. Load the data from the AWS S3 to Redshift using the Redshift Copy command.

copy employees

from 's3://test\_bucket/json/employee.json’

credentials 'aws\_iam\_role=arn:aws:iam::xxxxx:role/Redshift'

region 'eu-west-3'

json 's3://test\_bucket/jsonpath/employee\_jsonpath.json';

1. You will get a successful message when the above command finishes execution.
2. Use the below command to check the loaded data.

select \* from employee

Voila! You have successfully loaded the data from JSON to Redshift using the COPY command.