

Practicals With IBM Cloudant DB

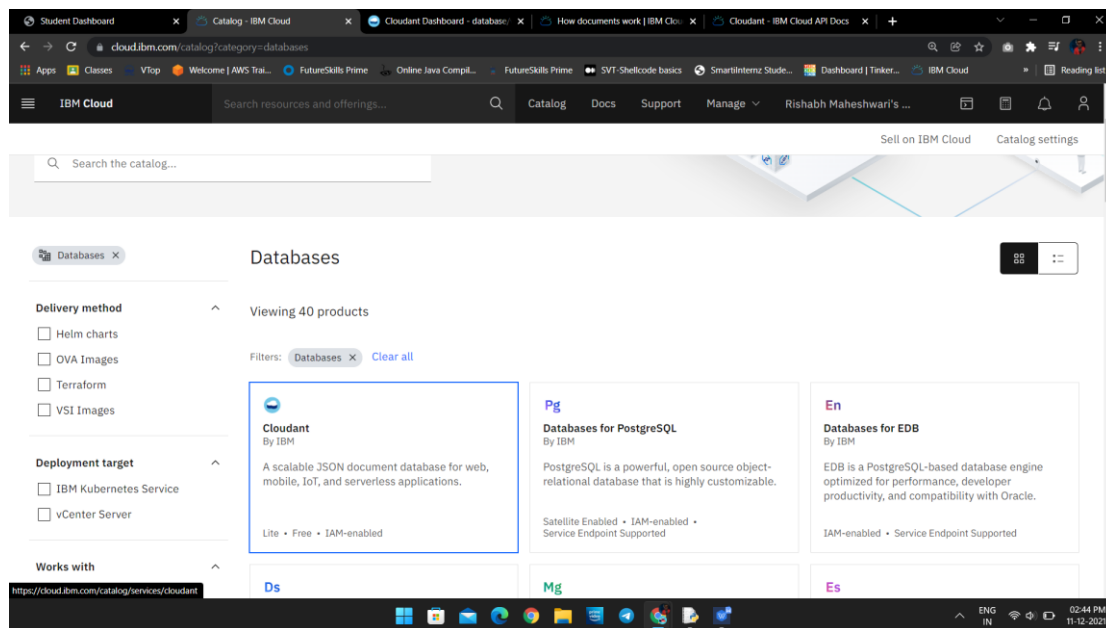
Rishabh Maheshwari

19BCY10145

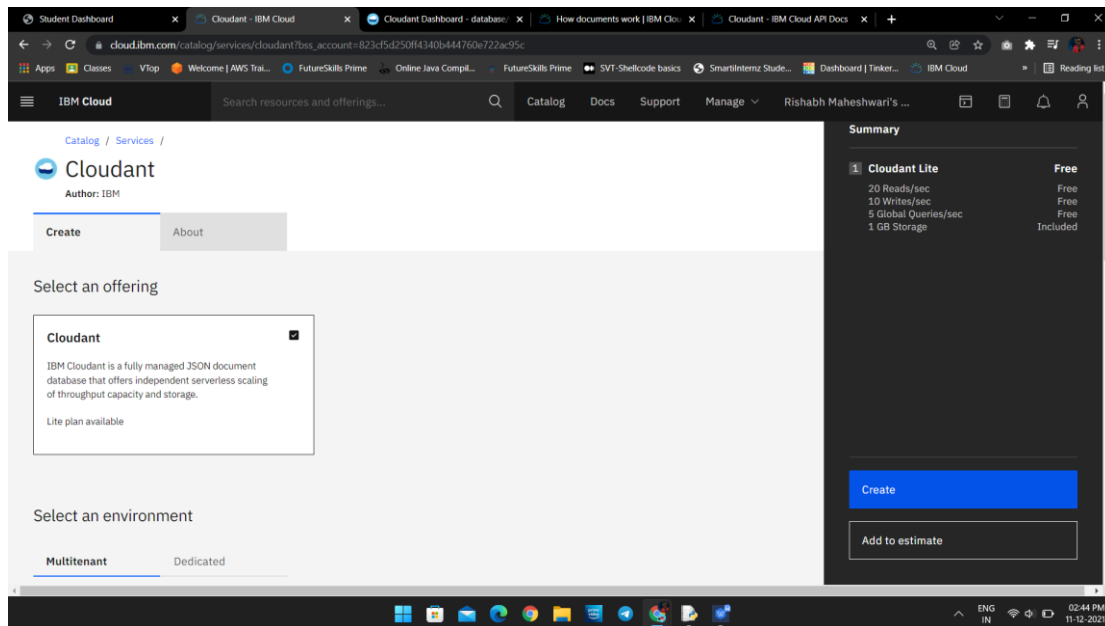
Explore IBM Cloudant DB, Create a database to store sensor data.

Store the Sensor data in the Cloudant Db and retrieve the data from the DB using Node-RED.

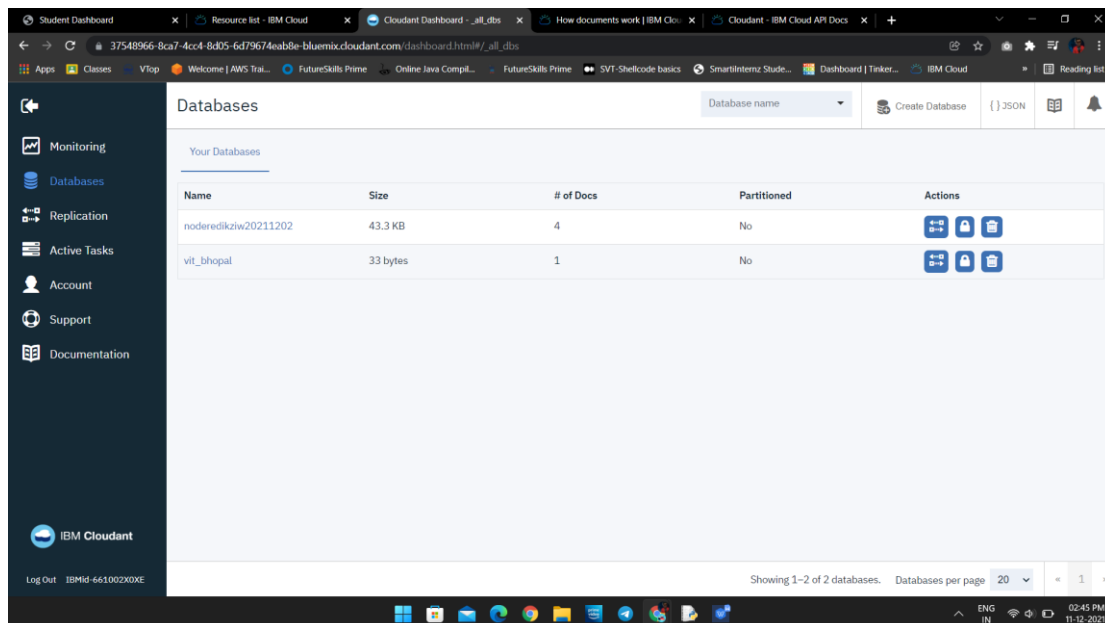
1. Open IBM Cloud and Search for Cloudant in Catalog after applying Databases as filter.



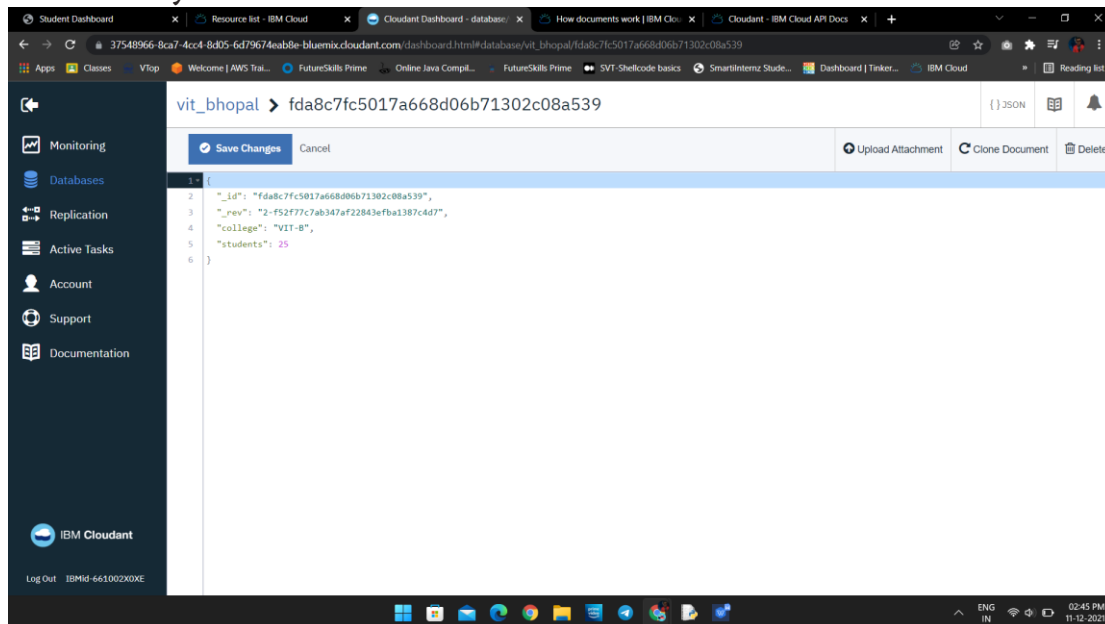
2. Click on Cloudbant Database service and Create the service by selecting Create.



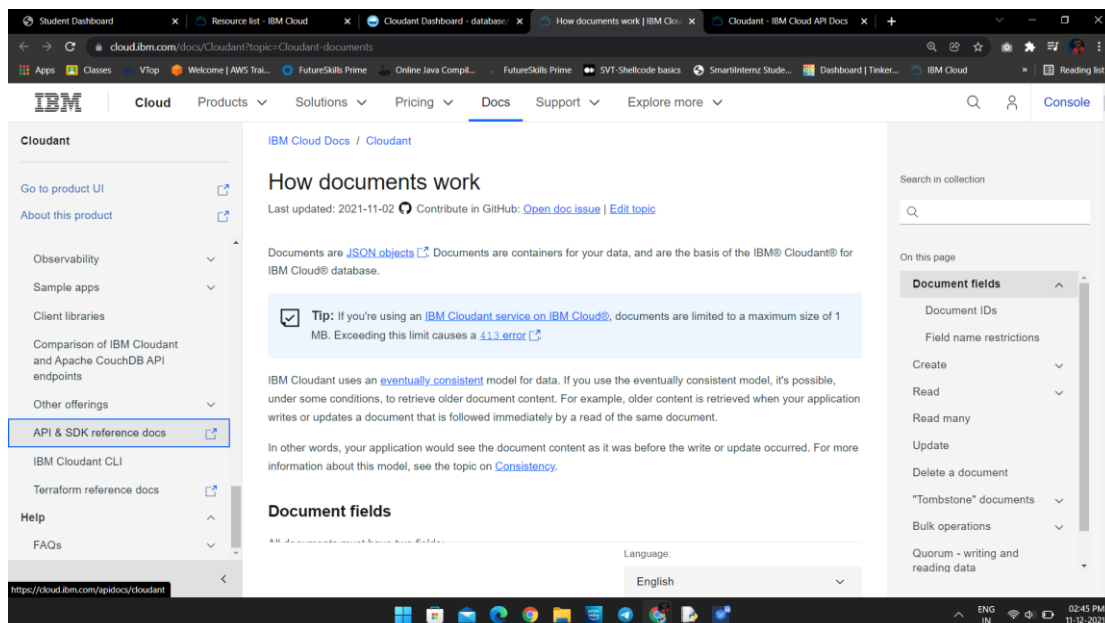
3. A default Database will be present (do not delete it). create new Database by selecting Create Databases.



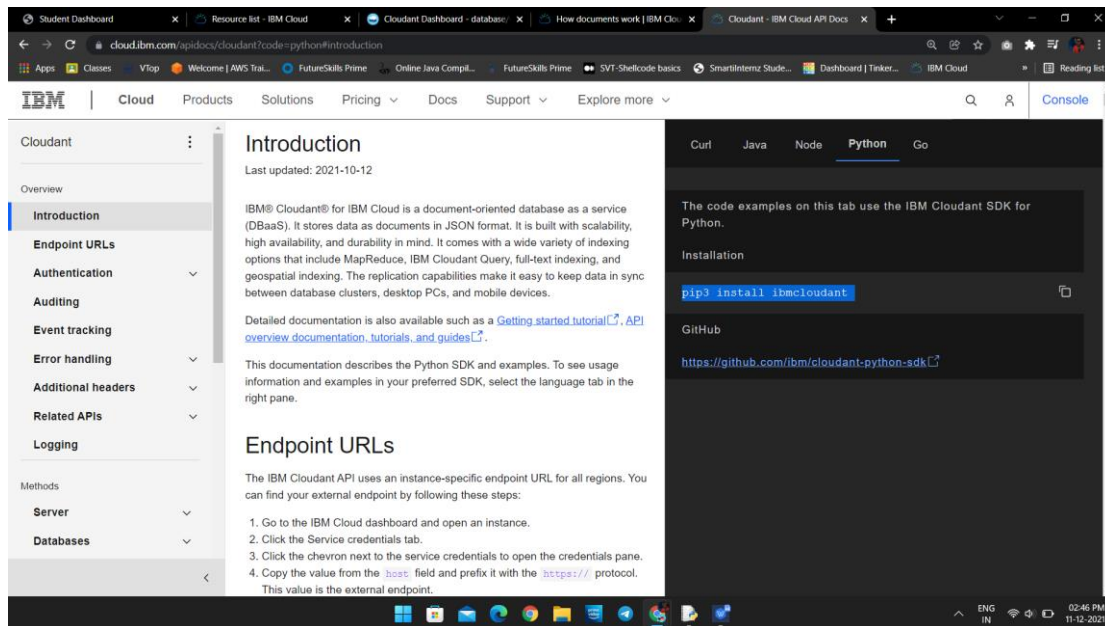
4. After creating the database a new field named “_rev” is created by itself.



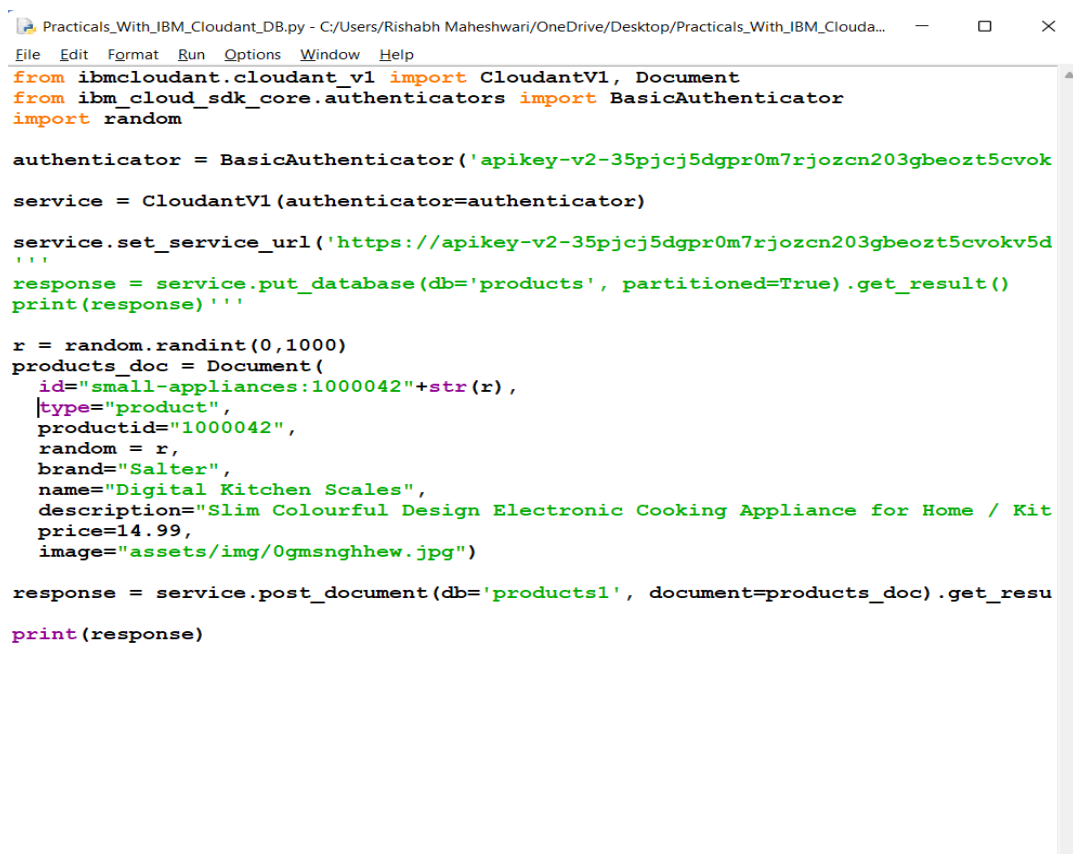
5. For sending data from device to cloud we use a python code. To get its format we have to search the documentations. So, click on the documentation icon and then search for API & SDK reference docs since, the data is sent using SDKs.



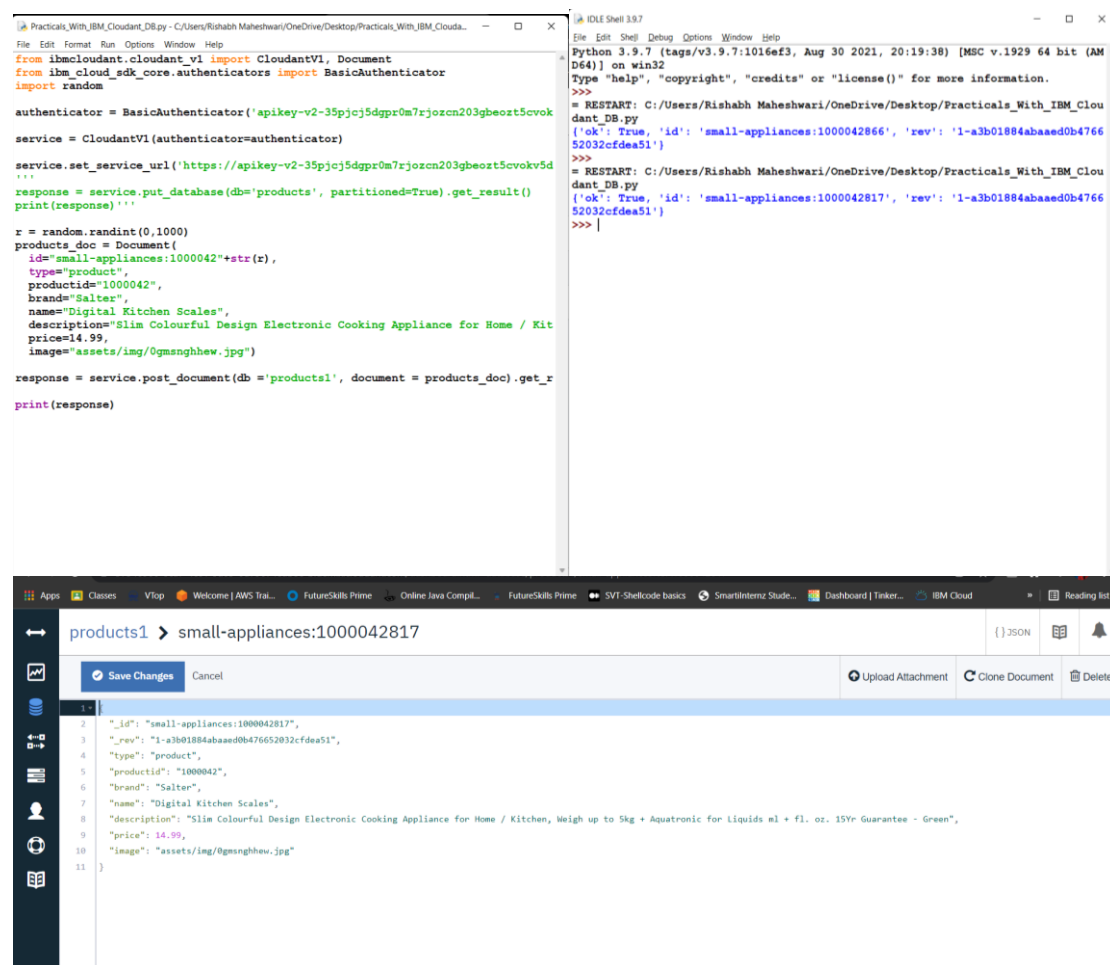
6. First we have to install the ibmcloudant directory using pip installation.



7. Now, to Create and store Sensor Data the python code will be as shown below.



8. New Database named Product1 will be created and Data will be save under the ID name.



The screenshot displays a Python IDE on the left and the IBM Cloud dashboard on the right. The IDE shows a script that uses the IBM Cloud SDK to create a new database named 'products1' and insert a document. The document contains details about a 'Digital Kitchen Scales' product. The dashboard on the right shows the 'products1' database with a single document retrieved, displaying the same product details as the one inserted by the script.

```
from ibmcloudant.cloudant_v1 import CloudantV1, Document
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator
import random

authenticator = BasicAuthenticator({'apikey-v2-35pjcj5dgp0m7rjozcn203gbeozt5cvokv5d'})
service = CloudantV1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-35pjcj5dgp0m7rjozcn203gbeozt5cvokv5d@cloudant-1000042817.cloudant.net')
response = service.put_database(db='products1', partitioned=True).get_result()
print(response)

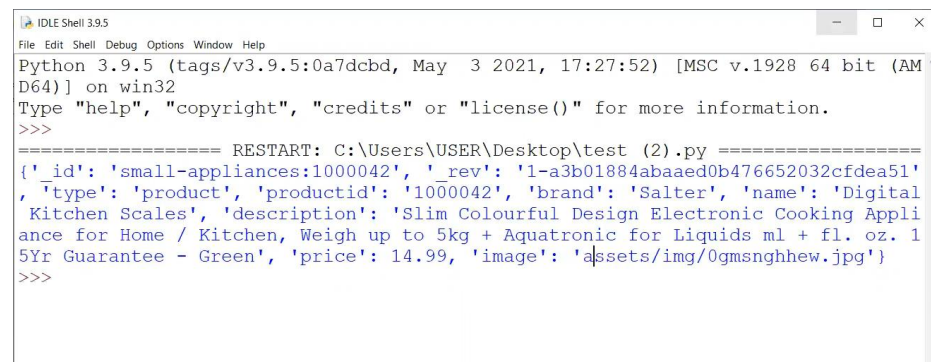
r = random.randint(0,1000)
products_doc = Document({
    'id': 'small-appliances:1000042'+str(r),
    'type': 'product',
    'productid': '1000042',
    'brand': 'Salter',
    'name': 'Digital Kitchen Scales',
    'description': 'Slim Colourful Design Electronic Cooking Appliance for Home / Kitchen, Weigh up to 5kg + Aquatronic for Liquids ml + fl. oz. 15Yr Guarantee - Green',
    'price': 14.99,
    'image': 'assets/img/0gmsnghew.jpg'
})

response = service.post_document(db='products1', document = products_doc).get_result()
print(response)
```

products1 > small-appliances:1000042817

```
{
  "_id": "small-appliances:1000042817",
  "_rev": "1-a3b01884abaaed0b476652032cfdea51",
  "type": "product",
  "productid": "1000042",
  "brand": "Salter",
  "name": "Digital Kitchen Scales",
  "description": "Slim Colourful Design Electronic Cooking Appliance for Home / Kitchen, Weigh up to 5kg + Aquatronic for Liquids ml + fl. oz. 15Yr Guarantee - Green",
  "price": 14.99,
  "image": "assets/img/0gmsnghew.jpg"
}
```

9. To retrieve the data using python code the Output would be as shown.

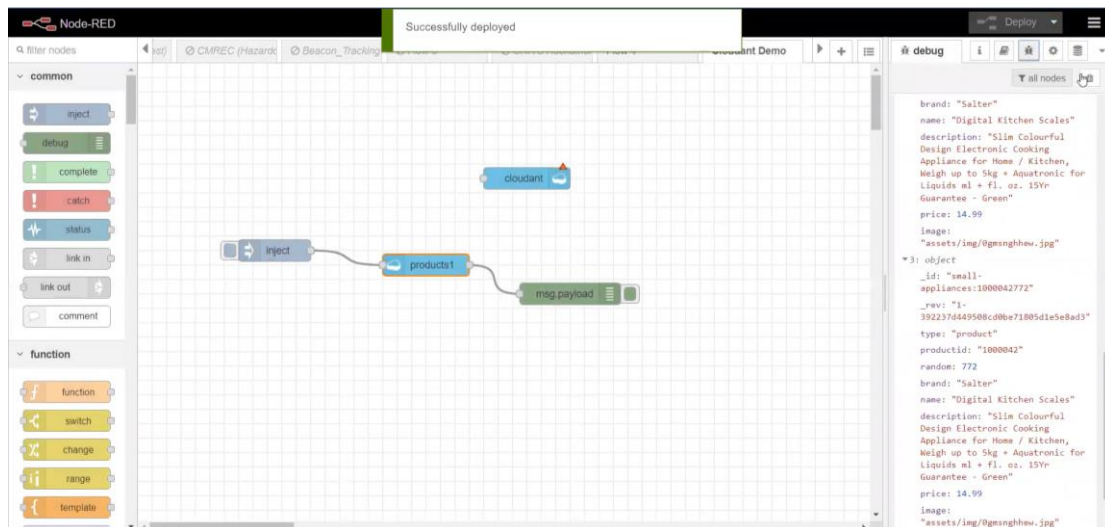


The screenshot shows a Python IDE window titled 'IDLE Shell 3.9.5'. It displays the output of a script that retrieves data from the 'products1' database. The output is a JSON object representing the product details, identical to the one shown in the previous screenshot.

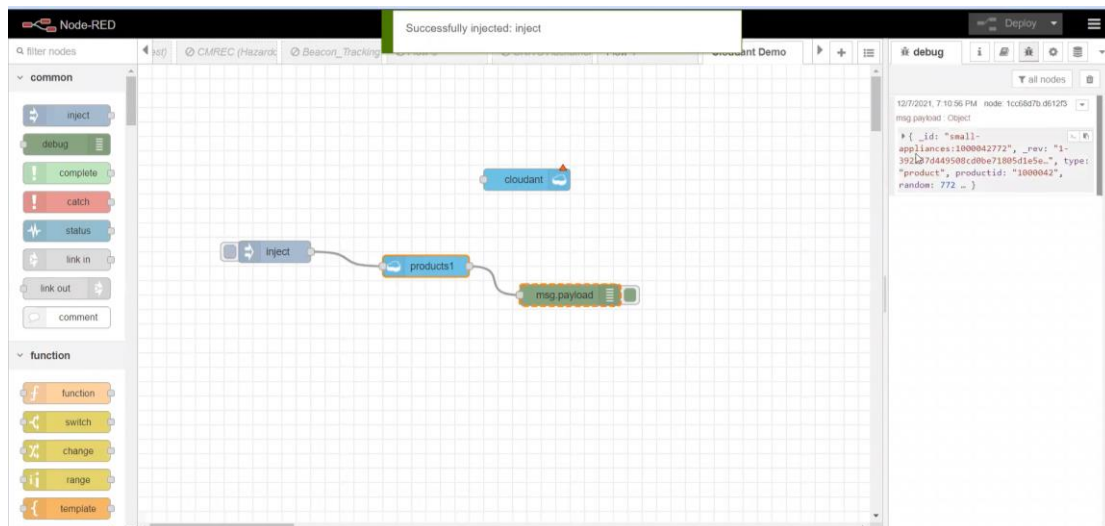
```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\USER\Desktop\test (2).py =====
{'_id': 'small-appliances:1000042', '_rev': '1-a3b01884abaaed0b476652032cfdea51', 'type': 'product', 'productid': '1000042', 'brand': 'Salter', 'name': 'Digital Kitchen Scales', 'description': 'Slim Colourful Design Electronic Cooking Appliance for Home / Kitchen, Weigh up to 5kg + Aquatronic for Liquids ml + fl. oz. 15Yr Guarantee - Green', 'price': 14.99, 'image': 'assets/img/0gmsnghew.jpg'}
```

10. To Send and Retrieve Data using Node-RED create the following flow.

Deploy and the Retrieved Data will be visible in the debug Console.



11. Click the Inject button to inject and store data in cloud database.



12. Click the Inject Output button in the flow to Retrieve Data from Database.

