

# **CT UNIVERSITY**



## **FUNDAMENTALS OF CLOUD COMPUTING LAB**

**(CSE207-22)**

**Lab File**

**SUBMITTED TO:  
Mr. Robin Tyagi**

**SUBMITTED BY:  
Name: LAKSHMAN  
KUMAR DAS  
Roll No: 72211502  
Class: B.TECH(IBM-CS)  
SEC-B**

# INDEX

Sr. No	Name of Experiment	Date	Faculty Signature	Remarks
1	Create an IBM Cloudant instance on IBM cloud and perform the CRUD (create, retrieve, update and delete) operation with third party application like postman	07/04/2023		
2				
3				
4				
5				
6				
7				
8				
9				
10		'		





# Practical 3

**Definition:** Create a database in JSON format and share its information on a third party application like postman.

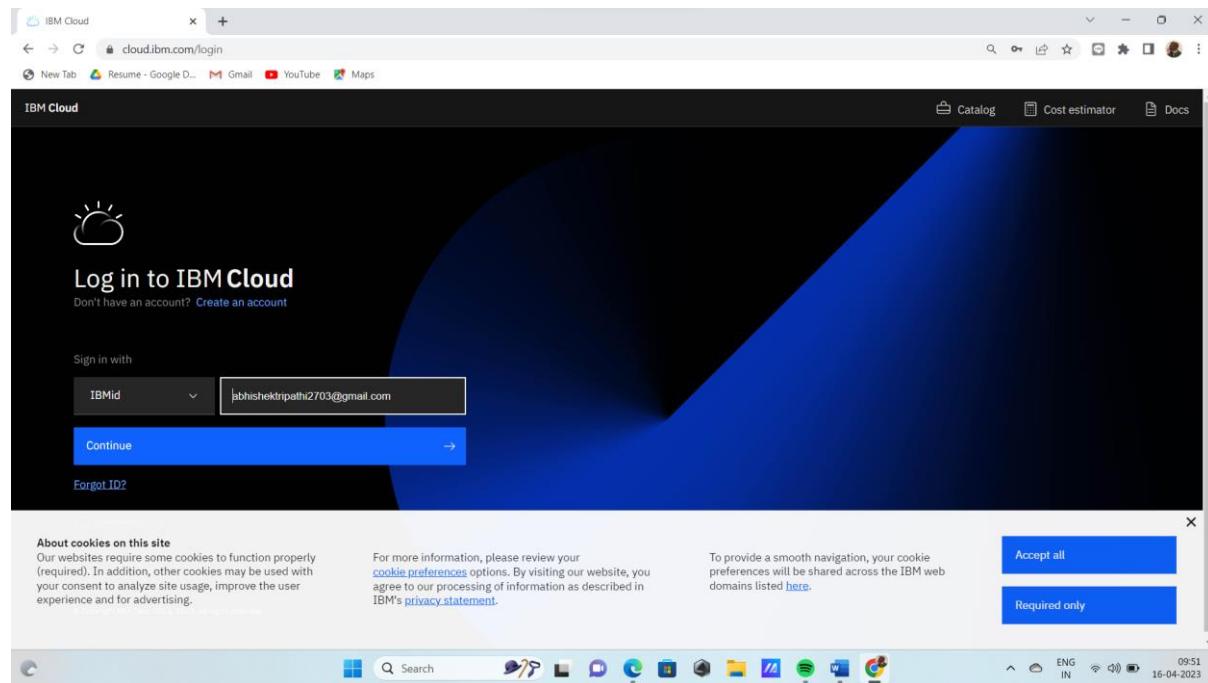
**Aim:** Create an IBM Cloudant instance on IBM cloud and perform the CRUD (create, retrieve, update and delete) operation with third party application like postman.

**Working:** using ibm cloud login to create, retrieve, update and delete

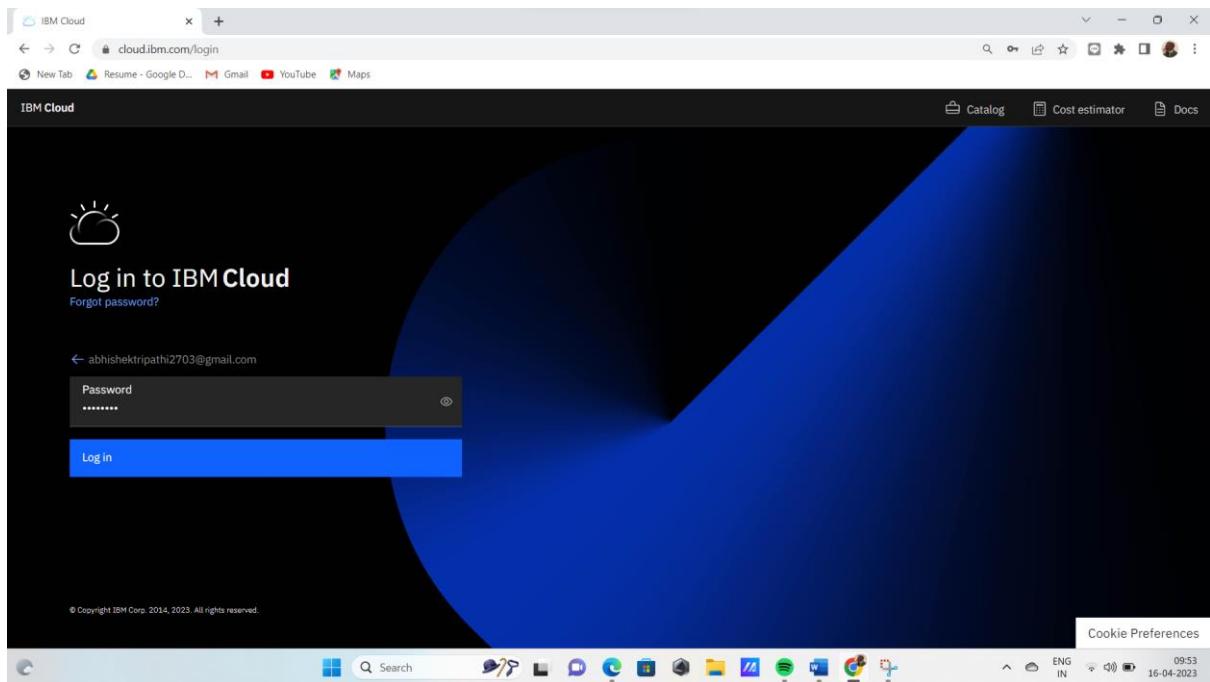
**Outcomes/Learning:** At the end of this practical we will be able to share our database to third party application like postman.

**Required Tool:** IBM CLOUD, AN INTERNET CONNECTION,CREDENTIALS.

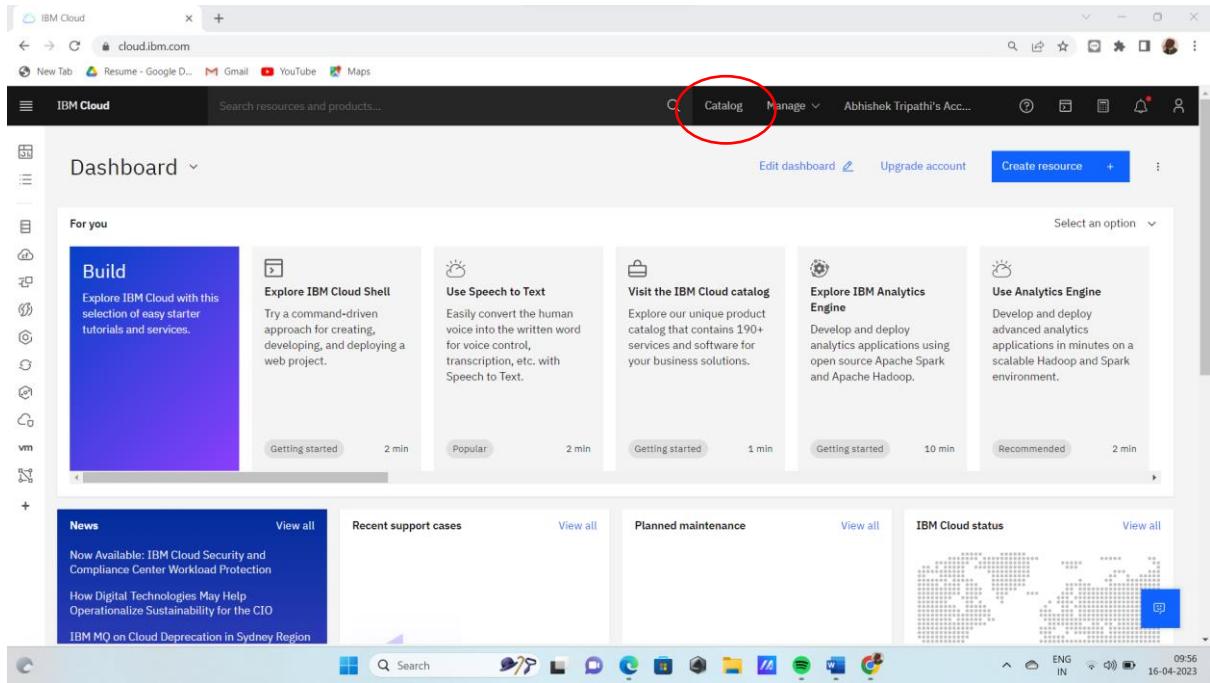
STEP 1 – On a browser search for ‘IBM CLOUD LOGIN’



STEP – Fill all the required credentials to sign in.



STEP - 3 First we need to create a data base . To do so click on catalog and click on 'DATABASES' then select option 'CLOUDANT'



The screenshot shows the IBM Cloud Catalog homepage. At the top, there's a navigation bar with links for 'Catalog', 'Manage', and 'Abhishek Tripathi's Acc...'. Below the navigation is a search bar labeled 'Search resources and products...'. The main area is titled 'Catalog' and features a search bar 'Search the catalog...'. On the right side, there's a 3D-style illustration of various cloud-based services like databases, storage, and networking. Below the illustration, there's a section titled 'Viewing 200 products' with a dropdown menu set to 'Alphabetically'. A sidebar on the left lists categories: Compute (32), Containers (10), Networking (29), Storage (19), AI / Machine Learning (18), Analytics (9), Blockchain (1), Databases (22), and Developer tools (11). The 'Databases (22)' link is highlighted with a red box.

This screenshot shows the 'Databases' category page within the IBM Cloud Catalog. The URL in the address bar is 'cloud.ibm.com/catalog?category=databases'. The main content area is titled 'Databases' and shows 'Viewing 22 products'. A yellow box highlights the first product card for 'Cloudant' by IBM. The card includes a brief description: 'A scalable JSON document database for web, mobile, IoT, and serverless applications.' Below the card, there are filter options: 'Filters: Databases X Clear all'. To the right of the card, there are other database offerings: 'Pg' (PostgreSQL), 'En' (EDB), 'Ds' (DataStax), and 'Mg' (MongoDB). The bottom of the screen shows a Windows taskbar with various pinned icons.

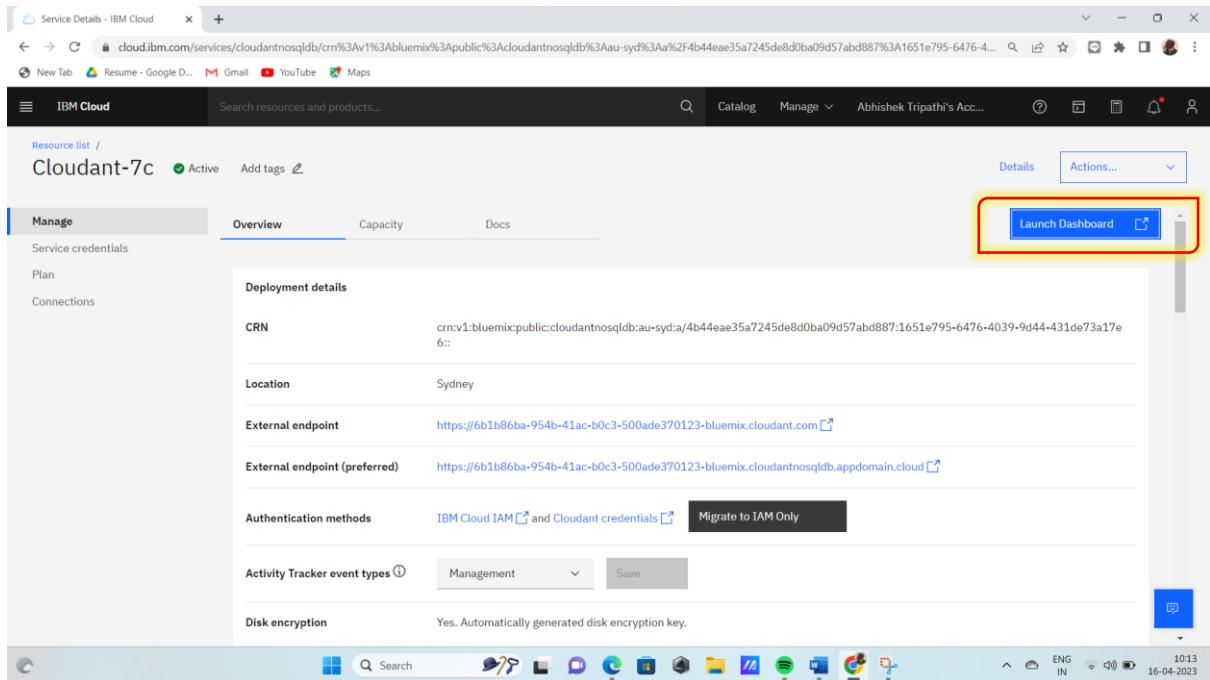
STEP – 4 After clicking on cloudant fill the required fields as shown and click on create.

The screenshot shows the IBM Cloud Catalog interface. In the center, there's a summary for the 'Cloudant Lite' plan, which is listed as 'Free'. It includes details like 20 Reads/sec, 10 Writes/sec, 5 Global Queries/sec, and 1 GB Storage Included. Below this, there are two plan options: 'Lite' (selected) and 'Standard'. The 'Lite' plan is described as having full functionality for development and evaluation with a set capacity, and it's noted that only one Lite plan instance per account is allowed. The 'Standard' plan is described as providing granular control over provisioned throughput capacity allocated. A note indicates that the 'Standard' plan starts at \$86.20/month. On the right side, there's a large blue 'Create' button, which is highlighted with a red box. Below it are 'Add to estimate' and a copy icon. At the bottom of the catalog page, there's a 'Cost calculator' link.

STEP -5 after creating cloudant instance go back to resource list and click on databases . there you will see new database created with name 'cloudant-4q'

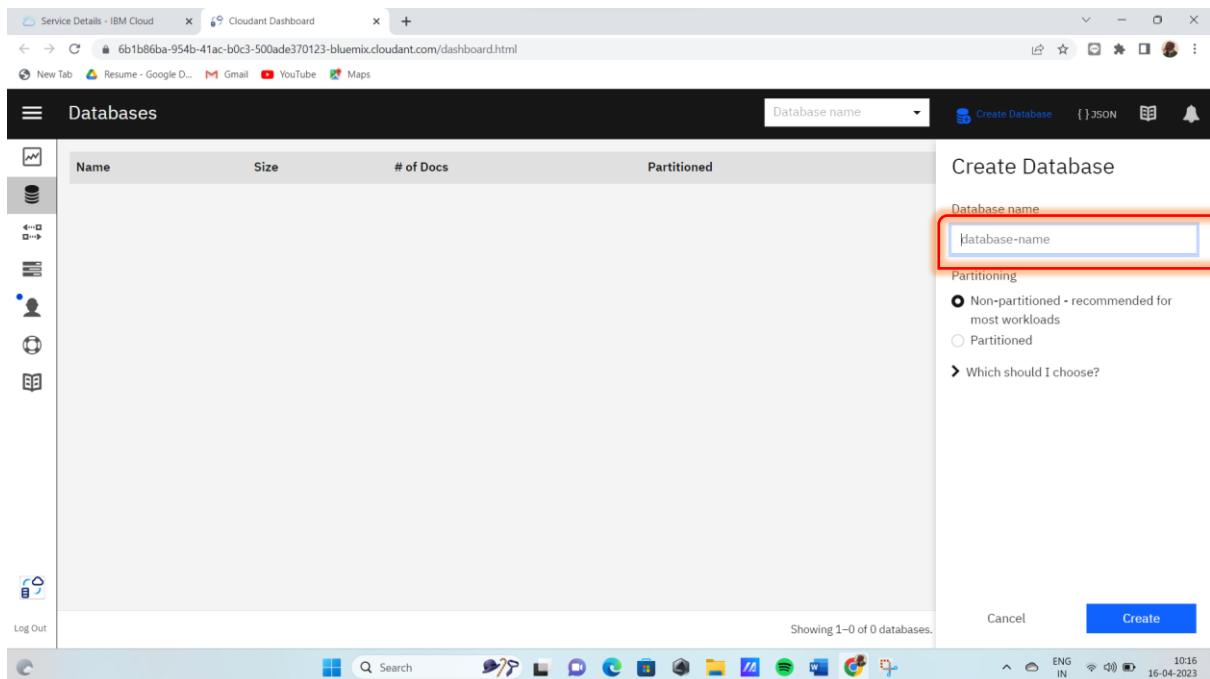
The screenshot shows the IBM Cloud Resource list. On the left, there's a sidebar with various service categories like Networking, Storage, Converged Infrastructure, Enterprise applications, AI / Machine Learning, Analytics, Blockchain, and Databases. The 'Databases' section is expanded, and within it, a single entry for 'Cloudant-7c' is listed. This entry includes columns for Name (Cloudant-7c), Group (Default), Location (Sydney), Product (Cloudant), Status (Provision in prog...), and Tags. The entire row for 'Cloudant-7c' is highlighted with a red box. The top navigation bar shows the URL as cloud.ibm.com/resources and the user as Abhishek Tripathi's Acc... The bottom taskbar shows the date as 16-04-2023 and the time as 10:05.

## STEP 6- Click on the database.



The screenshot shows the 'Service Details - IBM Cloud' interface for a Cloudant service named 'Cloudant-7c'. The 'Overview' tab is selected. On the right side, there is a 'Launch Dashboard' button, which is highlighted with a red box. Other visible details include the CRN, location (Sydney), and external endpoints.

STEP – 7 now click on ‘LAUNCH DASHBOARD’ to create document.give database name- ‘lakshman’



The screenshot shows the 'Cloudant Dashboard' interface. On the right, there is a 'Create Database' form. The 'Database name' field is highlighted with a red box. The form also includes sections for 'Partitioning' (with options for Non-partitioned and Partitioned) and a note about choosing a partition type.

Databases

Name	Size	# of Docs	Partitioned

Create Database

Database name:

Partitioning:

- Non-partitioned - recommended for most workloads
- Partitioned

► Which should I choose?

Cancel Create

STEP – 8 click on create and write in **JSON** format inside that document and remove the rev and save changes.

lakshman > 72211502

Save Changes Cancel

```

1 [
2   "_id": "72211502",
3   "_rev": "1-3f1f51be247315fa49e0391afb2b1e39",
4   "first name": "Lucky",
5   "last name": "Das",
6   "gender": "male",
7   "contact no.": "9334991692"
8 ]

```

Upload Attachment Clone Document Delete

Log Out

The screenshot shows the Cloudant Dashboard interface. On the left, there's a sidebar with icons for Query, Permissions, Changes, and Design Documents. The main area is titled 'lakshman' and shows 'All Documents'. A table displays one document entry:

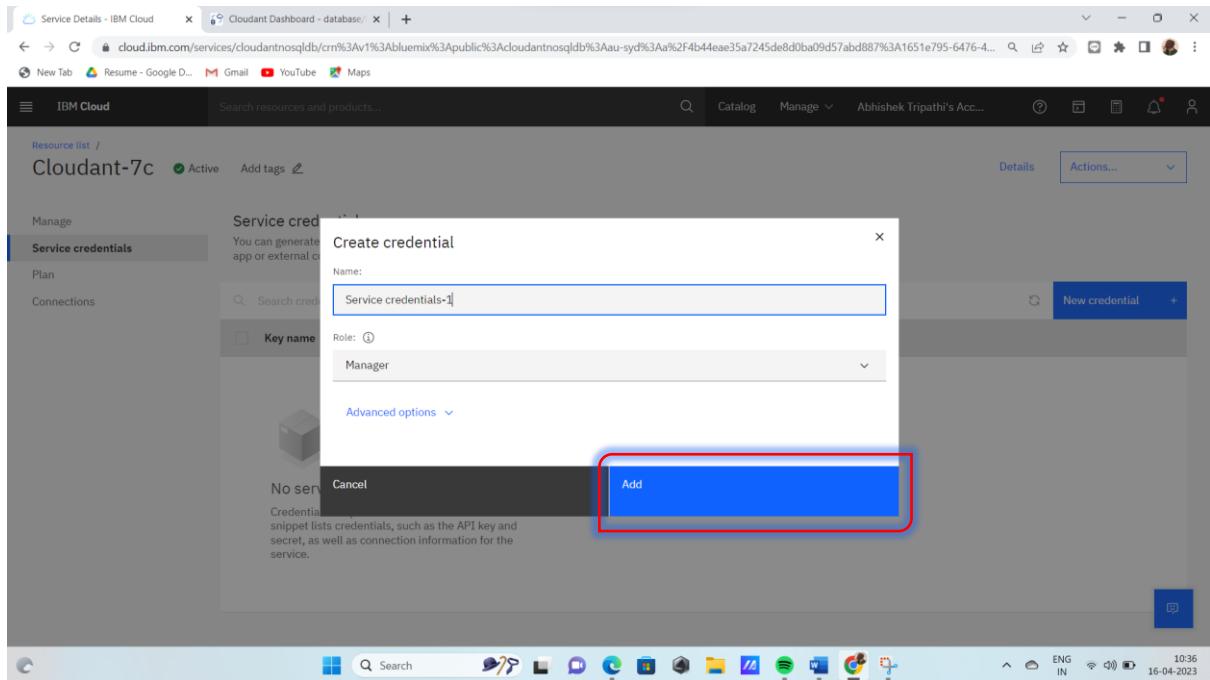
id	key	value
72211502	72211502	{"rev": "2-4c0bb69f4e01980936390591a62cb2a2"}

At the bottom right of the dashboard, it says 'Showing document 1 - 1. Documents per page: 20'. The system status bar at the bottom indicates 'ENG IN' and the date '16-04-2023'.

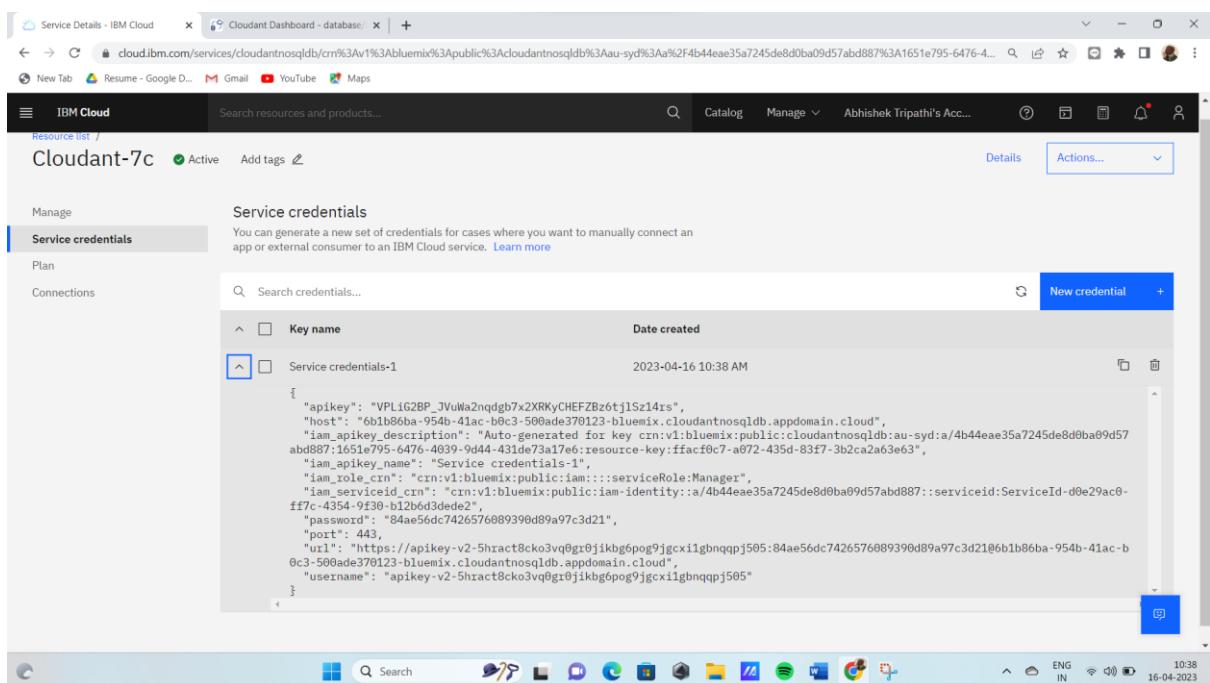
STEP – 9 click on service credentials.

The screenshot shows the 'Service Details - IBM Cloud' page for a service named 'Cloudant-7c'. The 'Service credentials' section is highlighted with a red box. It contains a sub-section titled 'Service credentials' with the following text: 'You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service.' Below this, there's a table with columns 'Key name' and 'Date created'. A 'New credential' button is located at the top right of this table area. The system status bar at the bottom indicates 'ENG IN' and the date '16-04-2023'.

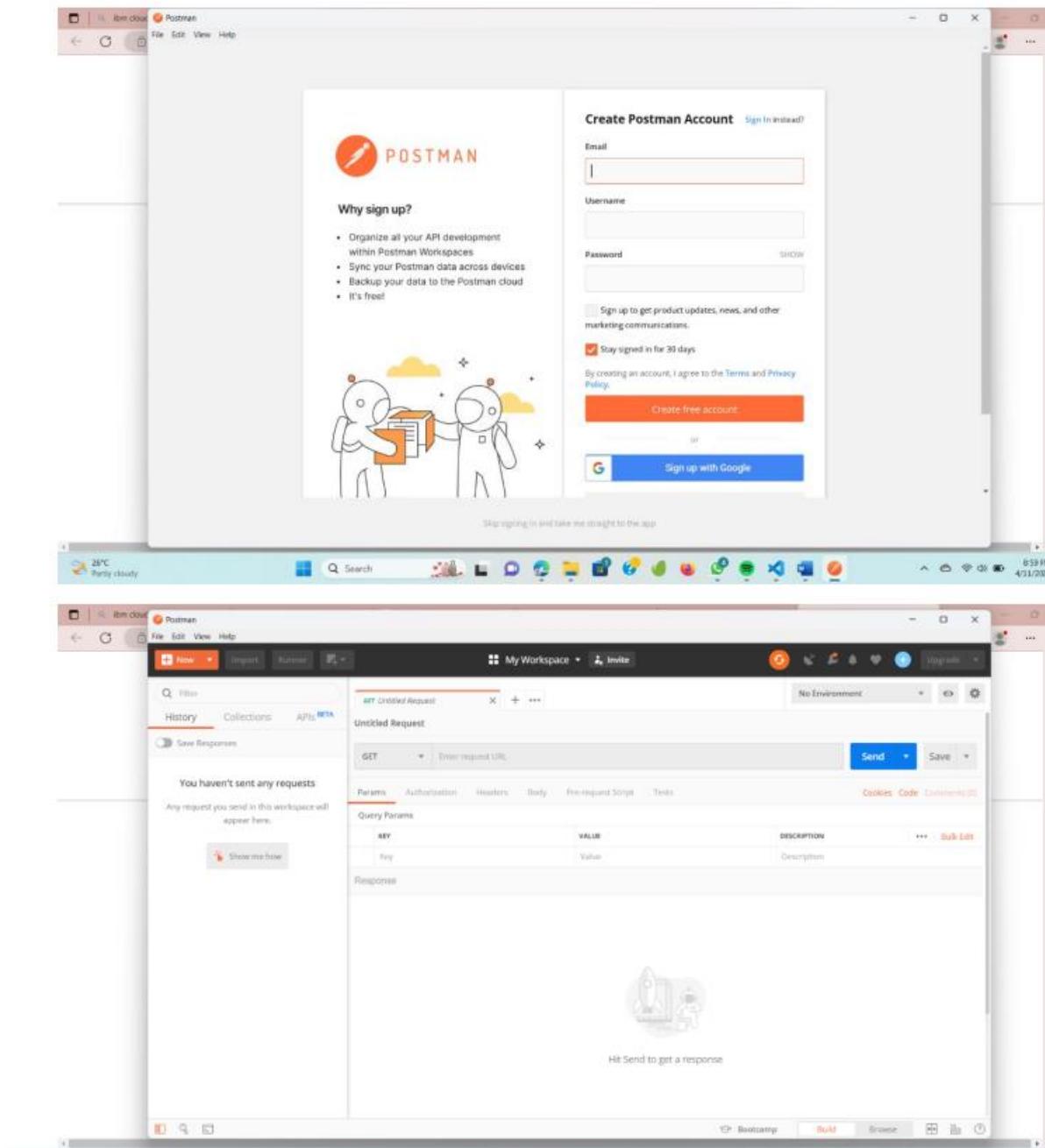
STEP – 10 create new service credential and add.



STEP – 11 Expand and we will get the url and api key required further.



STEP – 12 Now install a software named 'POSTMAN' and sign in.



## Create Access token on Postman

STEP -1 For api key click on service credentials on ibm cloudant and search for it as follows.

The screenshot shows the 'Service Details - IBM Cloud' window. In the top navigation bar, there are tabs for 'Service Details' and 'Cloudant Dashboard - database'. The URL in the address bar is [cloud.ibm.com/services/cloudantnosqldb/crn%3Av1%3Abluemix%3Apublic%3Acloudantnosqldb%3Aau-syd%3Aa%2F4b44eae35a7245de8d0ba09d57abd887%3A1651e795-6476-4...](https://cloud.ibm.com/services/cloudantnosqldb/crn%3Av1%3Abluemix%3Apublic%3Acloudantnosqldb%3Aau-syd%3Aa%2F4b44eae35a7245de8d0ba09d57abd887%3A1651e795-6476-4...). The main content area is titled 'Cloudant-7c' and shows the 'Service credentials' tab selected. It displays a table with one row, 'Service credentials-1', which was created on '2023-04-16 10:38 AM'. The table includes columns for 'Key name' and 'Date created'. The 'Key name' column contains a JSON object representing the API key. The JSON content is as follows:

```

{
  "apikey": "VPLIG2BP_JVuWa2nqgb7x2RKyCHFFZBz6tj1Sz14rs",
  "host": "6b1b86ba-954b-41ac-b0c3-500ade370123-bluemix.cloudantnosqldb.appdomain.cloud",
  "iam_apikey_description": "Auto-generated for key crn:v1:bluemix:public:cloudantnosqldb:au-syd:a/4b44eae35a7245de8d0ba09d57abd887:1651e795-6476-4039-9d44-431de73a17e6;resource-key:ffacf0c7-a072-435d-83f7-3b2ca2a63e63",
  "iam_apikey_name": "Service credentials-1",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam:identity::a/4b44eae35a7245de8d0ba09d57abd887:serviceid:ServiceId-d0e29ac0-ff7c-4354-9f30-b12b6d3dede2",
  "password": "84ae56dc7426576689390d89a97c3d21",
  "port": 443,
  "url": "https://apikey-v2-5hract8cko3vqfgr0jikbg6pog9jgcxi1gbnqqpj505:84ae56dc7426576689390d89a97c3d21@6b1b86ba-954b-41ac-b0c3-500ade370123-bluemix.cloudantnosqldb.appdomain.cloud",
  "username": "apikey-v2-5hract8cko3vqfgr0jikbg6pog9jgcxi1gbnqqpj505"
}

```

Put this api key in the above window of postman and also put the key and value pairs as listed below-

FOR KEYS AND VALUES –

**-Key: grant\_type and value:urn:ibm:params:oauth:grant-type:apikey**

**-Key: response\_type and value: cloud\_iam**

**-Key: apikey, and the value for this field should include your Cloudant instance API key**

And click on send.

POST https://iam.cloud.ibm.com/identity/token?

Key	Value	Description
grant_type	urn:ibm:params:oauth:grant-type:apikey	
response_type	cloud_ibm	
apikey	VPLIG2BP_JVuWa2nqdgB7x2XRKyCHEFZBz6tjSz...	

**Send**

```

{
  "access_token": "eyJraWQiOiIyM0IzM0QwMTA4MzE1LCJhbGciOiIzSUJ1Ni9...",
  "token_type": "Bearer"
}
  
```

**Send**

After sending ,The access token is displayed copy this in notepad.

## Now Retrieve document in Postman

Select get method and copy the following link-

\$URL/\$DATABASE/\_all\_docs?include\_docs=true

Find for url in service credentials (cloudant)

Name of database- lakshman

After filling the above credentials click on send and the document will be displayed on the screen.

```

1 {
2   "total_rows": 1,
3   "offset": 0,
4   "rows": [
5     {
6       "id": "72211502",
7       "key": "72211502",
8       "value": {
9         "rev": "3-e14a9114d7c3b4fe2fc72222cc6c656"
10      }
11    }
12  ]
13}
  
```

# Create new document in cloudant using postman tool

send a POST request to \$URL/\$DATABASE with the document's JSON content in the request body.

Update the request URL to \$URL/\$DATABASE

Update the HTTP method to POST.

click the Body tab ,select the raw radio button, and then select the type as JSON (application/json)

Enter the following code as the content of the document:

```
{  
  "_id": "novel_004",  
  "name": "The Merry Adventures of Robin Hood",  
  "author": "Howard Pyle",  
  "year": 1883  
}
```

The screenshot shows the Postman application interface. On the left, there is a sidebar with options like Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The History section is currently selected. In the main workspace, there is a message: "You haven't sent any requests. Any request you send in this workspace will appear here." Below this, there is a "Show me how" link. The central area shows a POST request configuration for the URL `https://apikey-v2-5hract8cko3vq0gr0jikbg6pog9jgcx1gbnqqp|505:84ae56dc7426576089390d89a97c3d21@6b1b86ba-9...`. The "Body" tab is selected, and the "raw" radio button is selected under the "Content Type" dropdown, which is set to "JSON". The "Params", "Authorization", "Headers", "Pre-request Script", "Tests", and "Settings" tabs are also visible. At the bottom, there is a "Response" section with a message: "Click Send to get a response". The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

The screenshot shows the Postman interface with a POST request to the URL `https://apikey-v2-5hract8cko3vq0gr0jikbg6pog9jgctx1gbnqqpj505:84ae56dc7426576089390d89a97c3d21@6b1b86ba-9`. The request body is set to `JSON` and contains the following JSON data:

```

1
2   "_id": "72211502",
3   "first_name": "Lucky",
4   "last_name": "Das",
5   "gender": "male",
6   "contact_no": "2555632125"
7

```

The 'Send' button is highlighted with a red box.

The screenshot shows the Postman interface after sending the POST request. The response details are as follows:

- Status: 200 OK
- Time: 2.31 s
- Size: 1.04 KB

The response body is displayed in `Pretty` format:

```

1
2   "total_rows": 2,
3   "offset": 0,
4   "rows": [
5     {
6       "id": "72211502",
7       "key": "72211502",
8       "value": {
9         "rev": "4-5c805d8f38e0b51e8dcc724e9481dc63"
10      }
11    }
12  ]

```

# Retrieve some selected document in postman using ID

Set the request method to the GET method.

Set the request URL to `$URL/$DATABASE/_all_docs?include_docs=true`

url- from service credentials

database-lakshman

Select the Headers tab and add the following key and value

Key: Authorization.

Value: Bearer \$ACCESS\_TOKEN

And click on send.

The screenshot shows the Postman application interface. In the center, there is a request card for a POST method to the URL `https://apikey-v2-5hract8cko3vq0gr0jikbg6pog9jgcx1gbnqqp505:84ae56dc7426576089390d89a97c3d21@6b1b86ba-9`. The 'Headers' tab is selected, showing an 'Authorization' header with the value 'Bearer \$ACCESS\_TOKEN'. The 'Body' tab is also selected, displaying a JSON payload:

```
1 {
2   "_id": "72211502",
3   "first name": "Lucky",
4   "last name": "Das",
5   "gender": "male",
6   "contact no.": "2555632125"
7 }
```

Below the request card, the 'Test Results' section shows the response details: Status: 200 OK, Time: 2.31 s, Size: 1.04 KB. The response body is displayed in Pretty format:

```
1 {
2   "_id": "72211502",
3   "_rev": "1-1759fdfb1cd0efcc289e79f4ae99fef2",
4   "first name": "Ramesh",
5   "last name": "Kumar",
6   "gender": "male",
7   "contact no.": "2555625515"
8 }
```

## Update document in cloudant using postman tool

To update a document, send a PUT request to \$URL/\$DATABASE/\$DOCUMENT\_ID

with the updated document JSON content and the latest \_rev field.

Keep the request URL as \$URL/\$DATABASE/stud\_02.

Update the HTTP method to PUT.

update the request body with the following code, as shown in the following figure.

Replace \$REV with the \_rev value that you copied in the previous step

```
"_rev": "$REV",
```

The screenshot shows the Postman interface with a successful PUT request. The URL is `https://apikey-v2-5hract8cko3vq0gr0jikbg6pog9jgctx1gbnqqpj505:84ae56dc7426576089390d89a97c3d21@6b1b86ba-9...`. The response status is 201 Created, and the response body is:

```

1 {
2   "_rev": "1-b23654c07079b34dd6459a6204a5fdd0",
3   "name": "Oliver Twist"
4 }

```

# Delete document in cloudant using postman tool

To delete a document, issue a DELETE request for `$URL/$DATABASE/$DOCUMENT_ID?rev=$REV`

The screenshot shows the Postman interface with an Untitled Request. The method is set to DELETE. The URL field contains `https://iam.clicloud.net/apikey-v2-5hract8cko3vq0.../stud01?rev=1-b23654c07079b34dd6459a6204a5fdd0`. The response area shows a cartoon character holding a rocket.

Update `$DOCUMENT_ID` with the ID of the document of the lakshman document, which is stud01 to stud02.