Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

Aim : Cloudant database in IBM Cloud integration with a third-party application (Postman)

1. Install third party application postman on your system

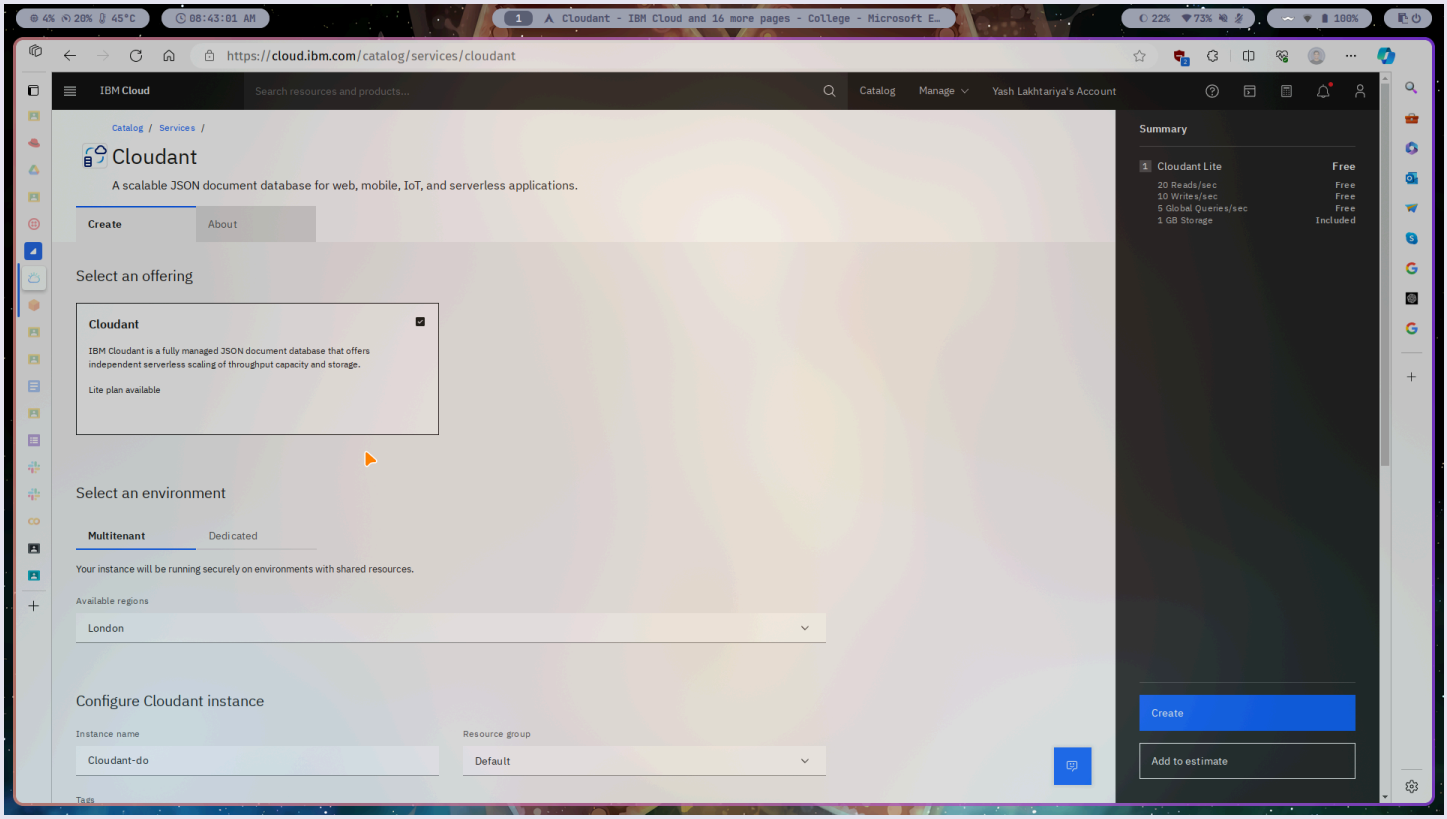
2. Create database for novel shop and document via creating an instance of Cloudant database.

4. Perform crud operations on database created using API's

5. Make PDF document for performed tasks

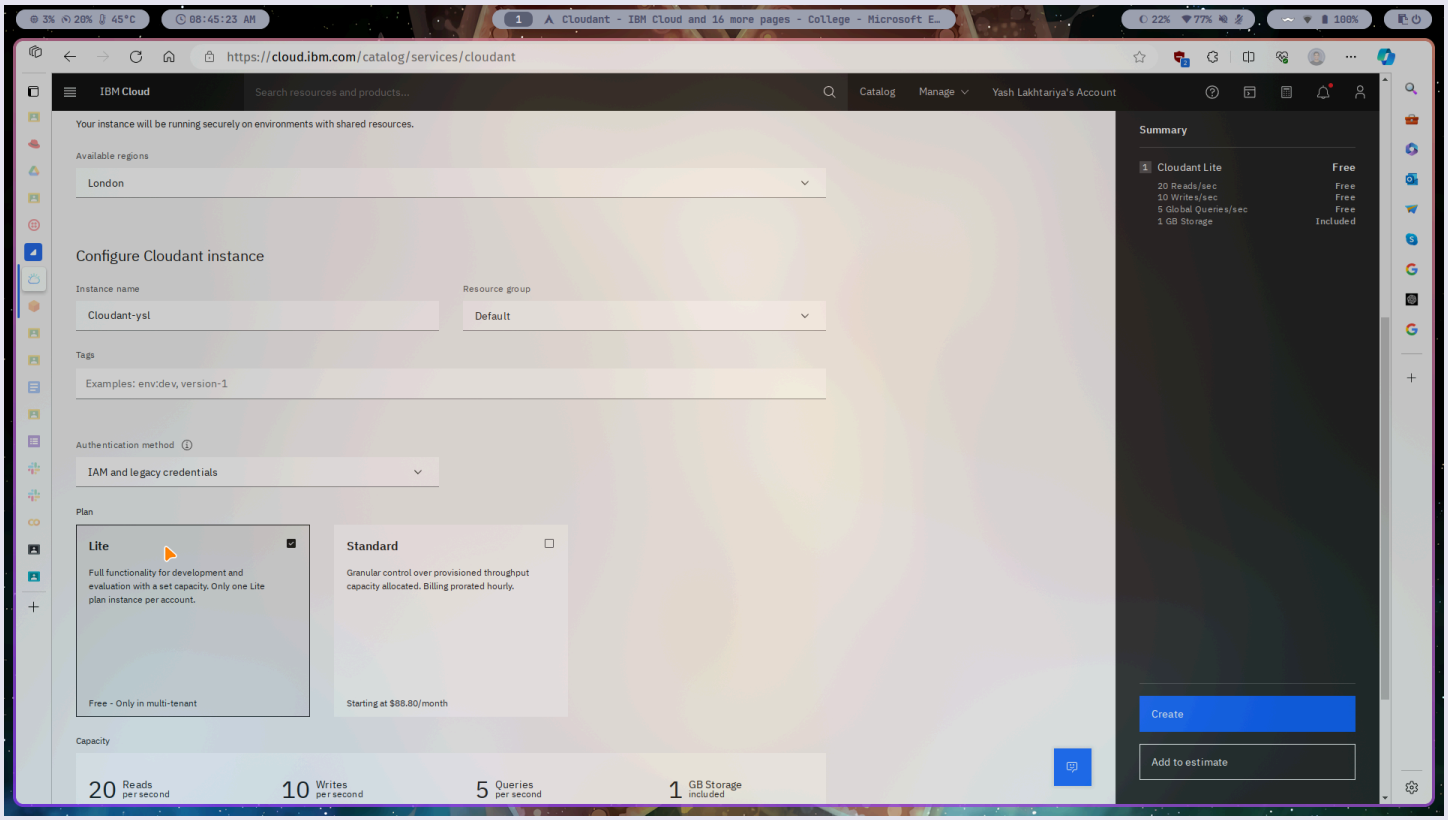
Steps and screenshots :

1. Create Cloudant service on IBM Cloud



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

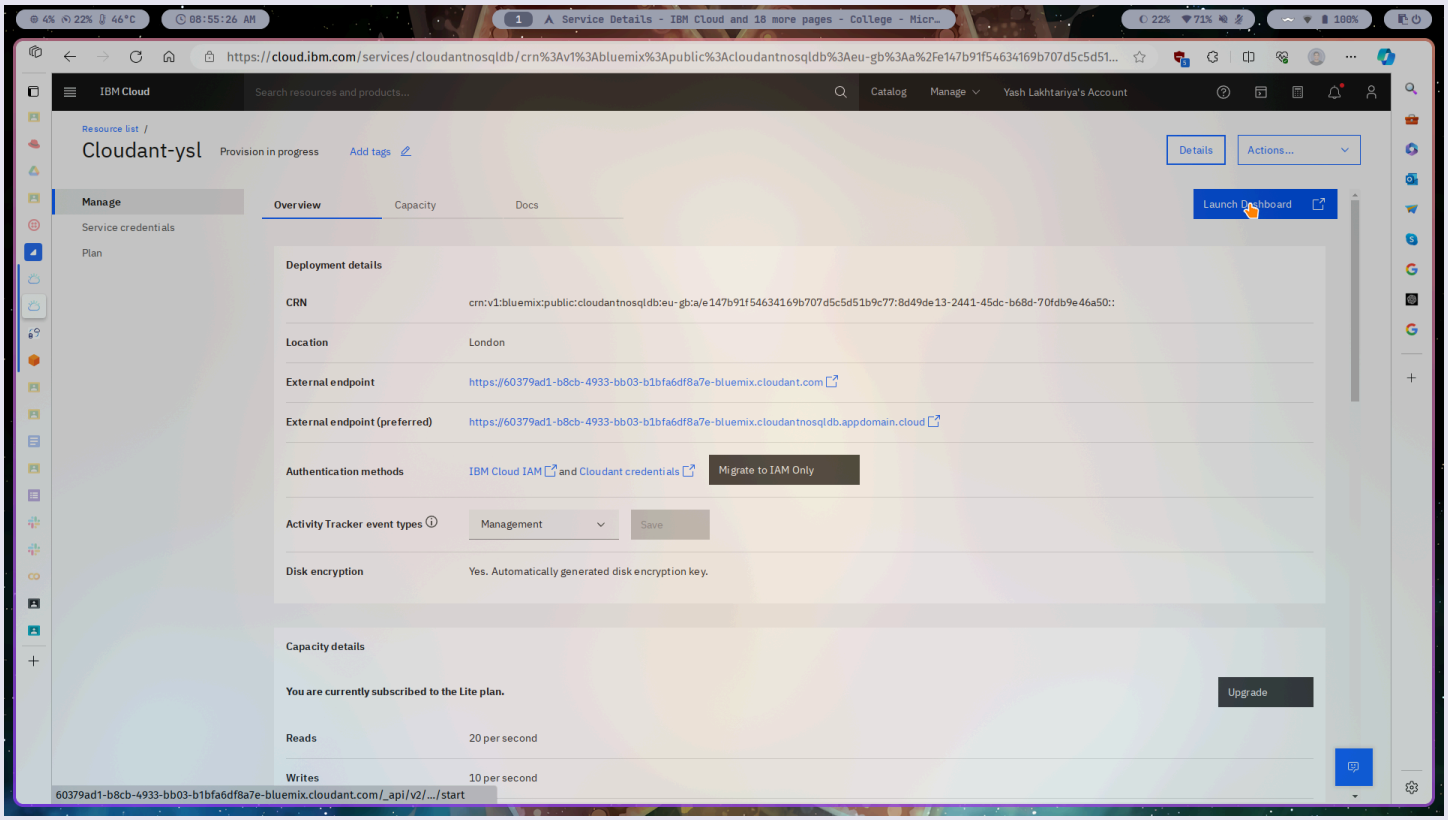
2. Select preferred region and choose ***IAM and Legacy Credentials*** as Authentication method and rest default options



(Here, to send API requests through third party app (postman), we need to choose IAM and legacy credentials, not IAM only)

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

3. Now, after the creation of service, the dashboard of cloudant can be launched via option on top right

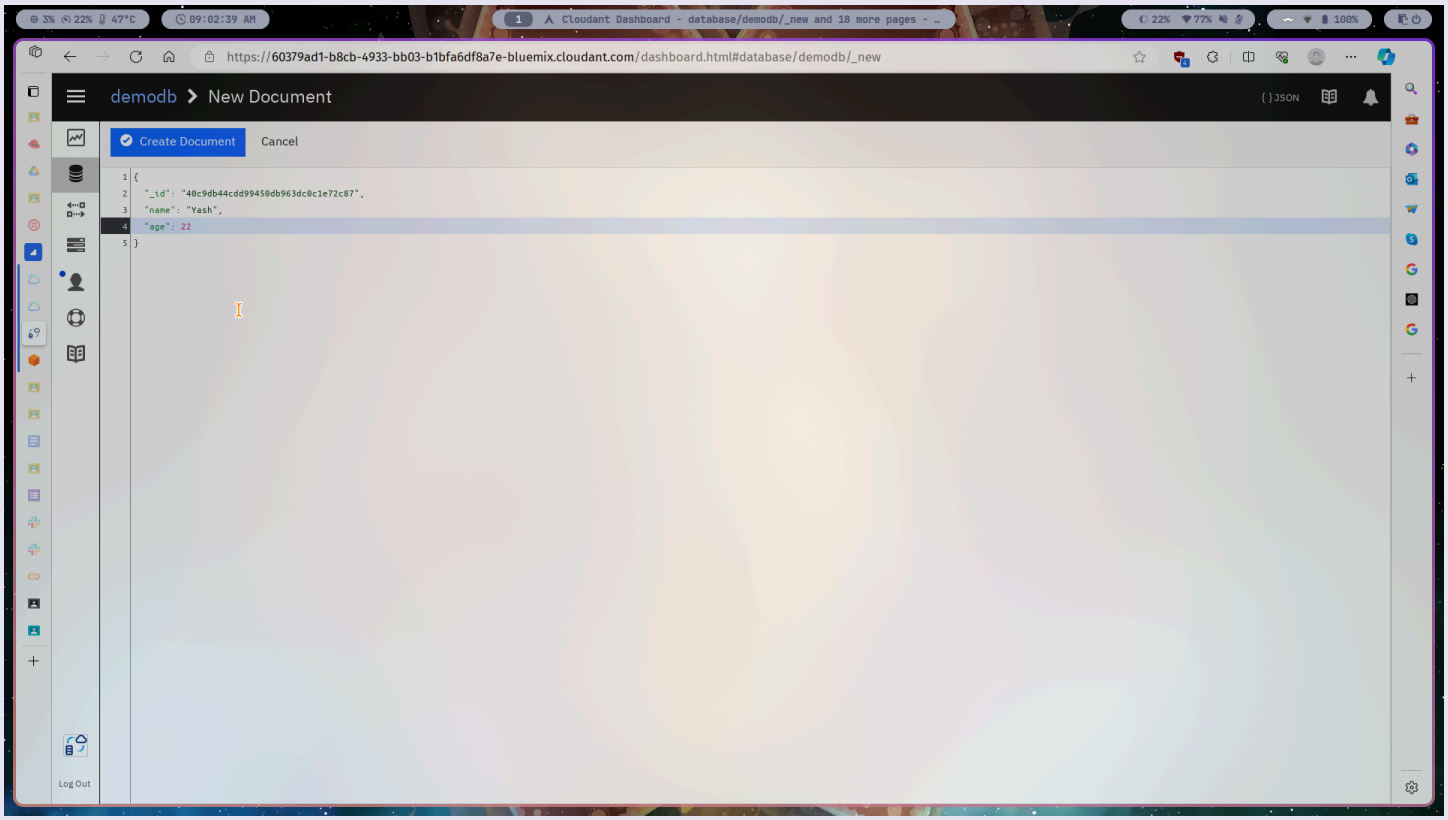


Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

4. On dashboard, in databases tab, create a database and specify the name. The partitioning option should be ***Non-partitioned***

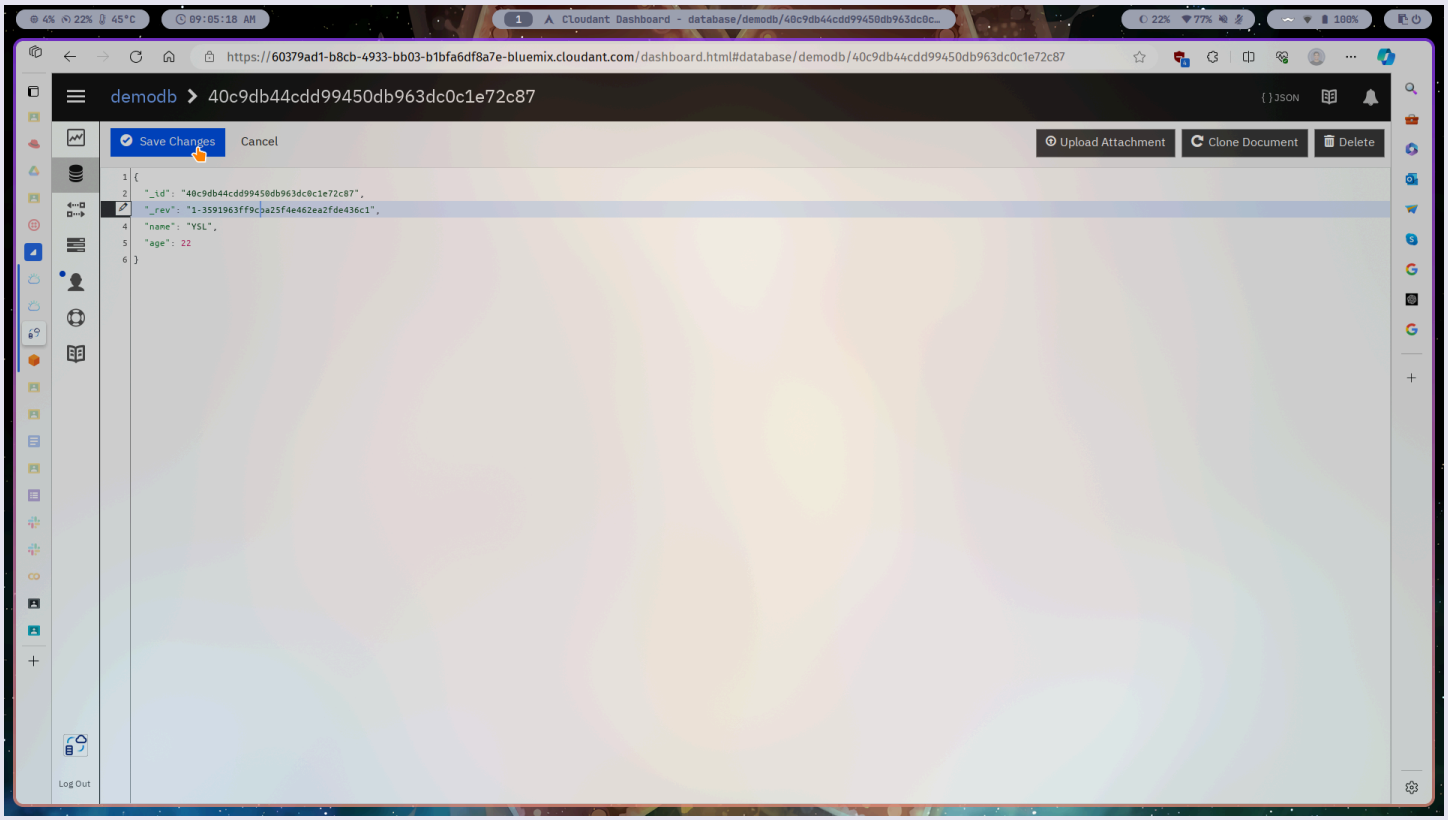
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

5. In the database created, create a document with some key-value pairs, as it is in JSON format



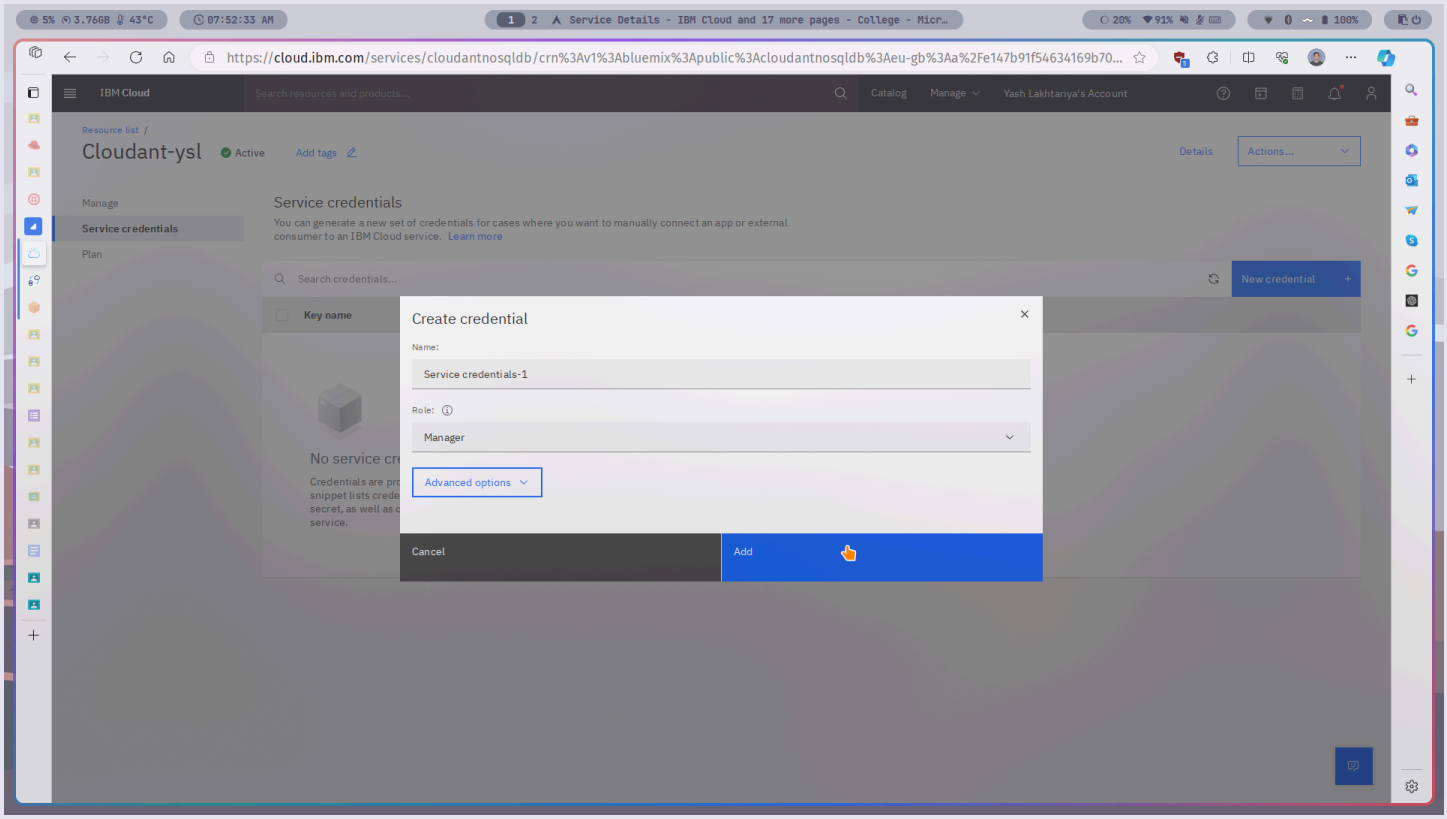
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

6. A revision ID is automatically generated which is used at many places, specifically, the first digit(s) states the number of times it is changed



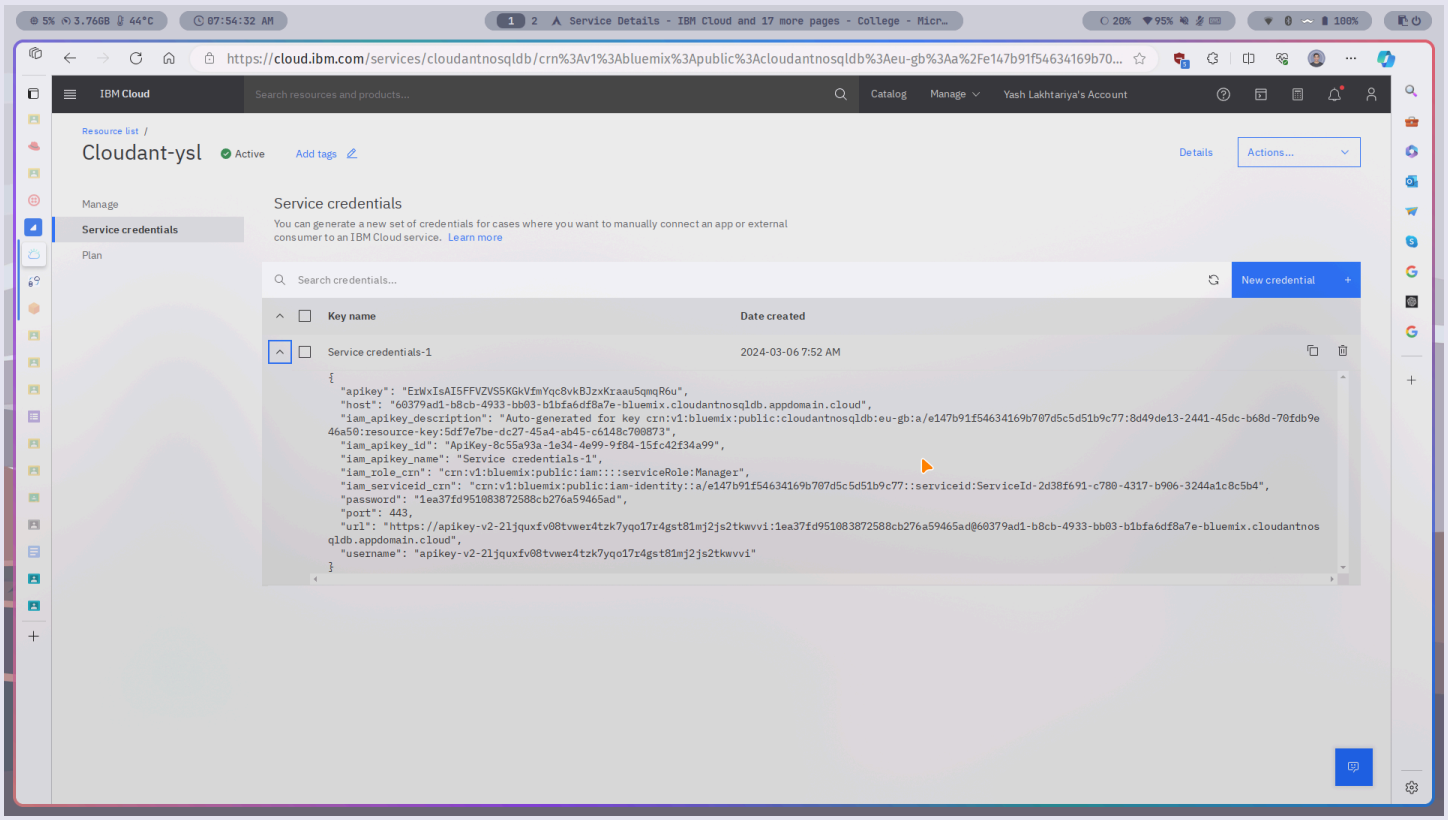
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

7. In IBM CLoudant service, now create credentials in Manage Credentials tab



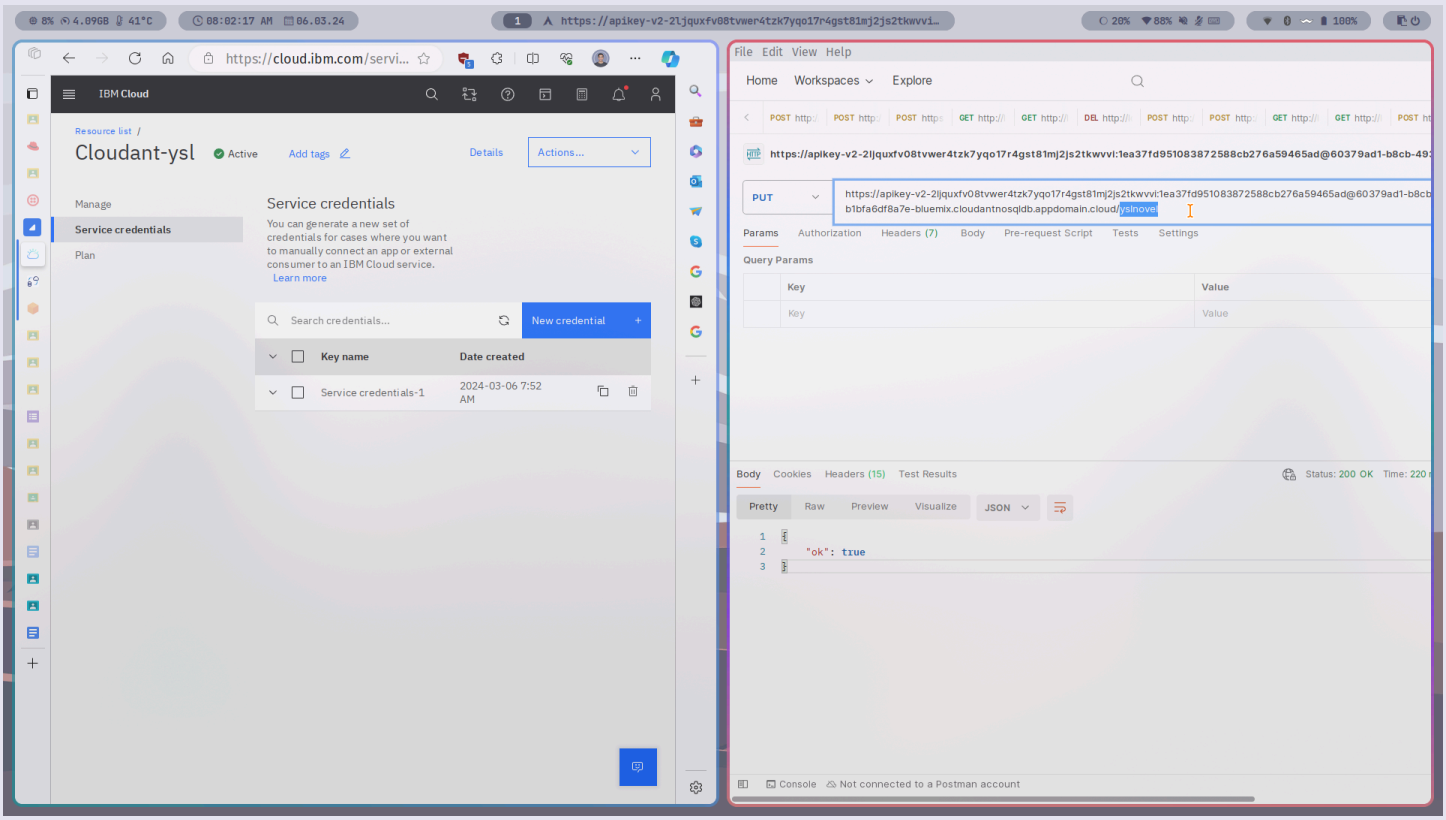
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

8. Check the credentials and copy the same, the URL with format username:password@host is mostly used to send requests via postman



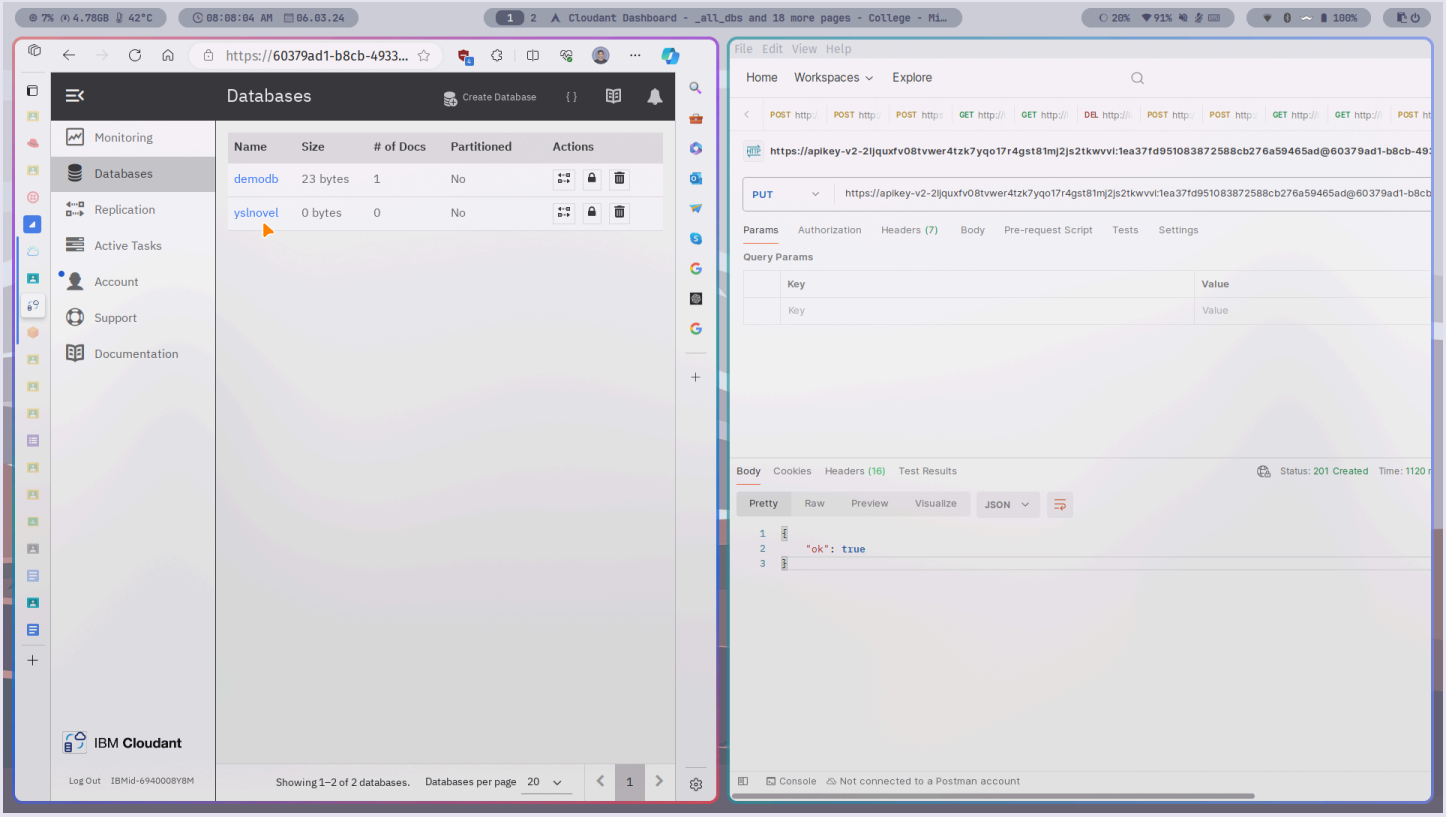
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

9. Now, using Postman application, send PUT request to the URL/databasename to create a new database



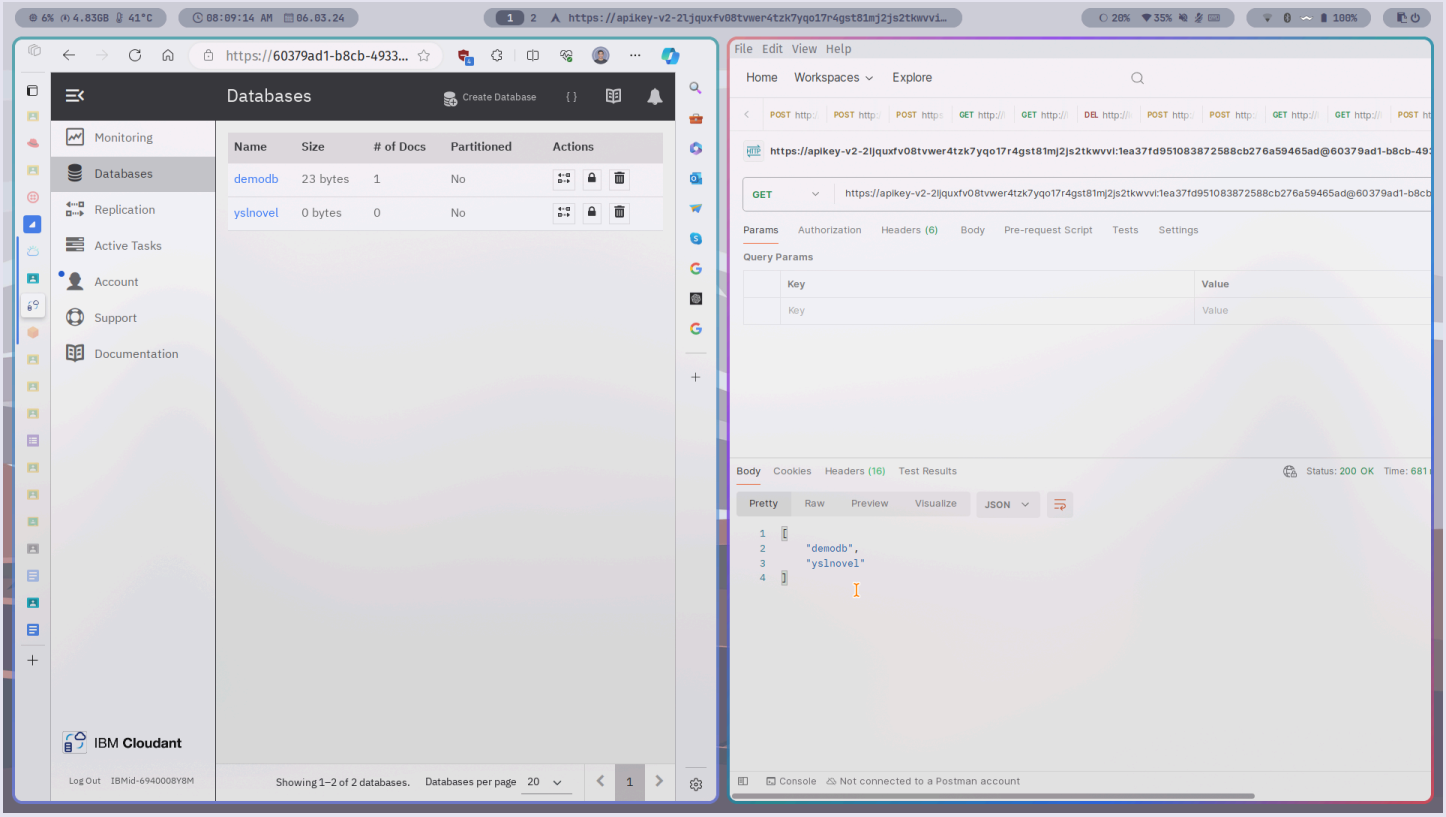
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

10. Check on the dashboard if the database is created successfully



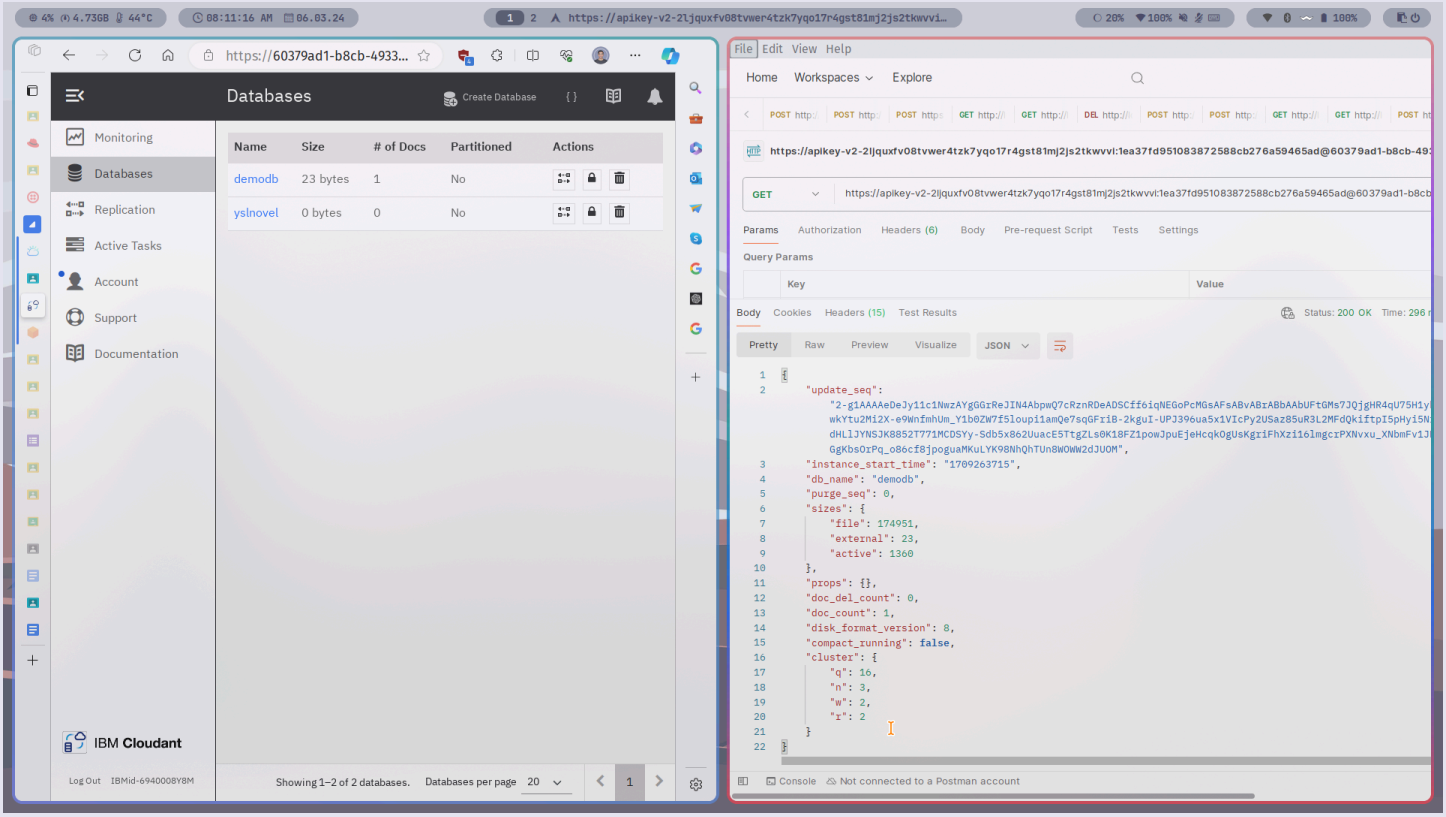
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

11. Now, make a GET request with URL/_all_dbs to get the list of existing databases



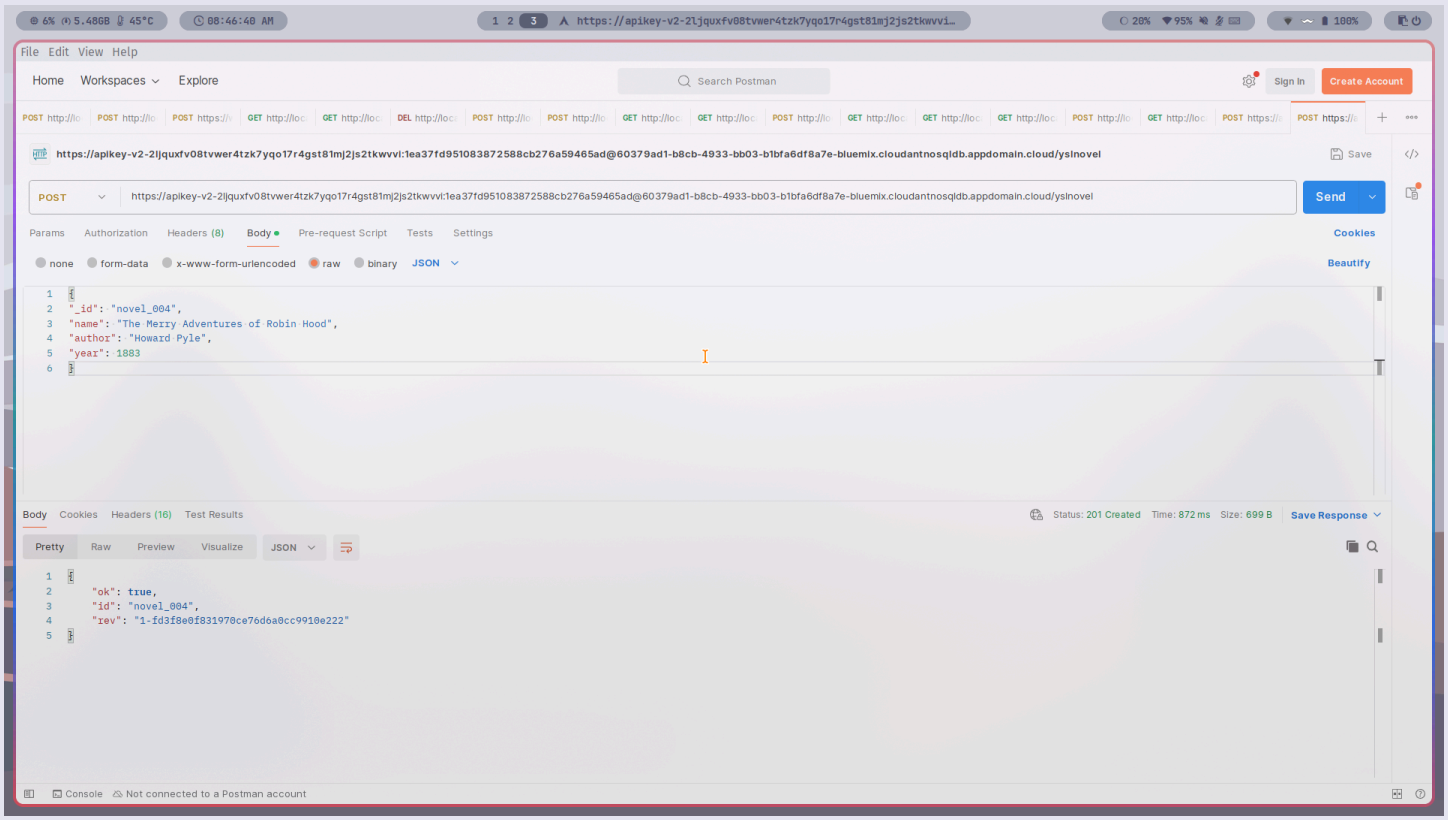
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

12. Now, make a GET request to the URL/database to get the specific database's details



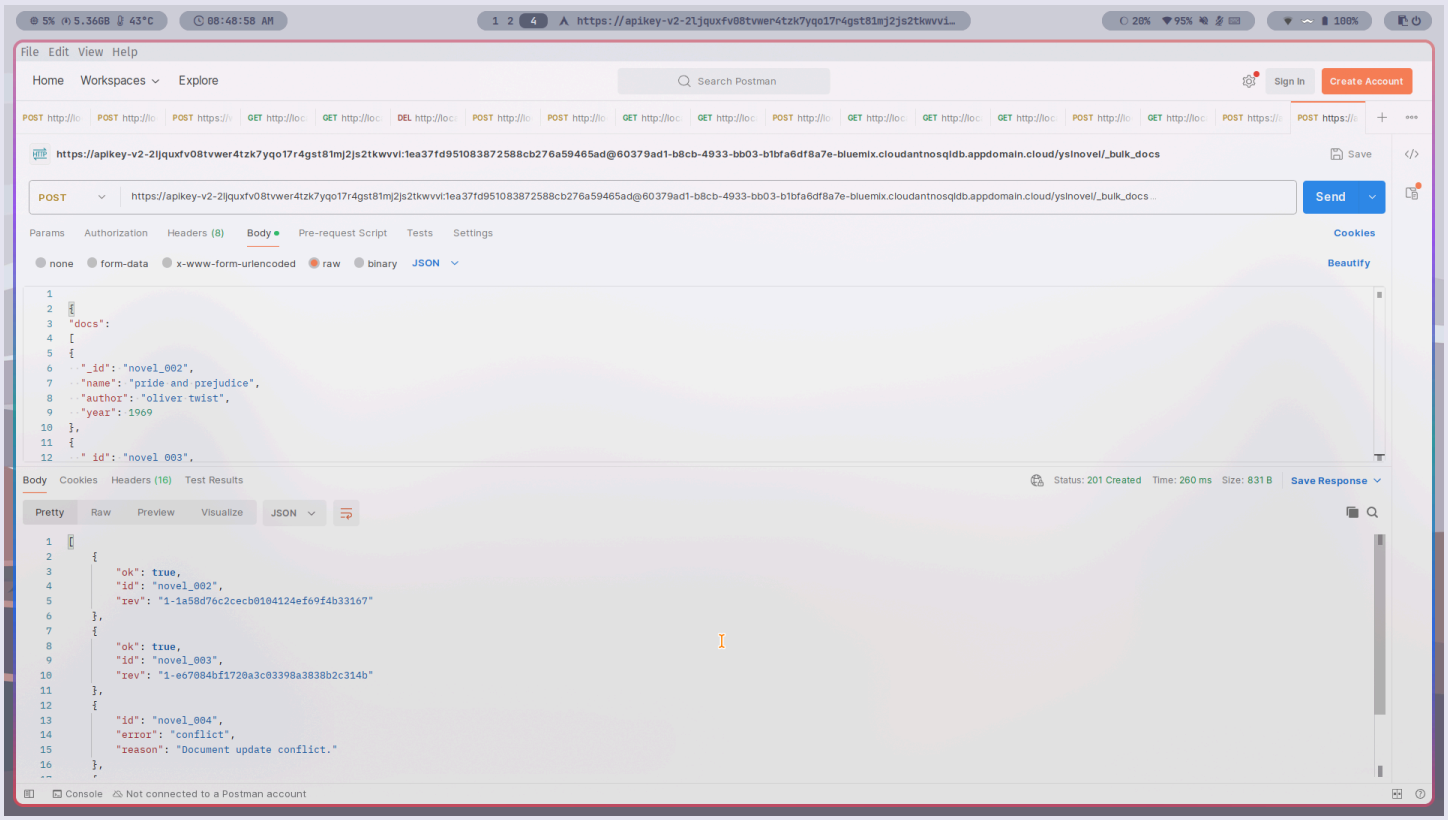
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

13. To add details, make a POST request to URL/database, with json details in body



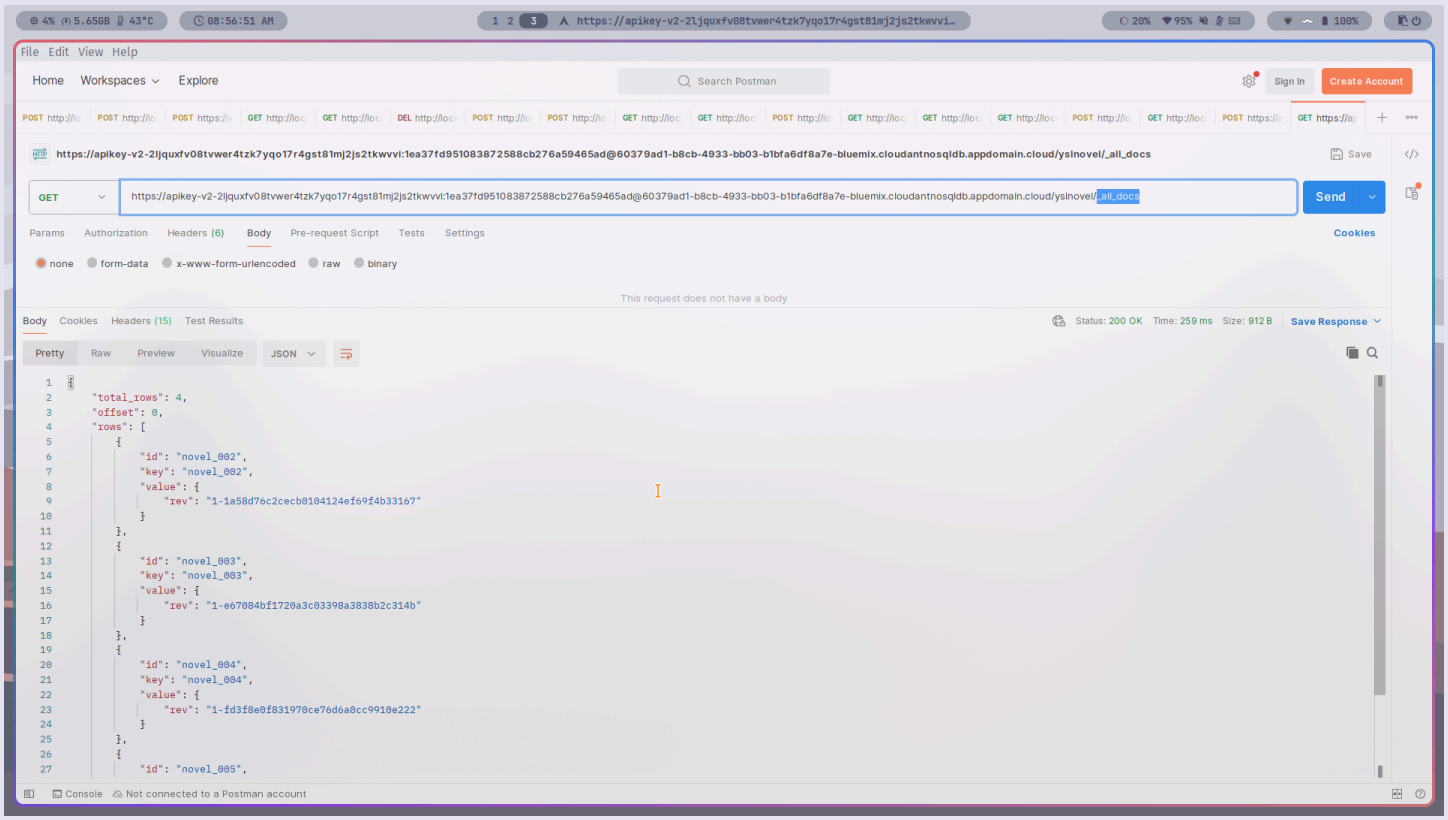
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

14. _bulk_docs can be added after database name for dealing with all docs in the database.



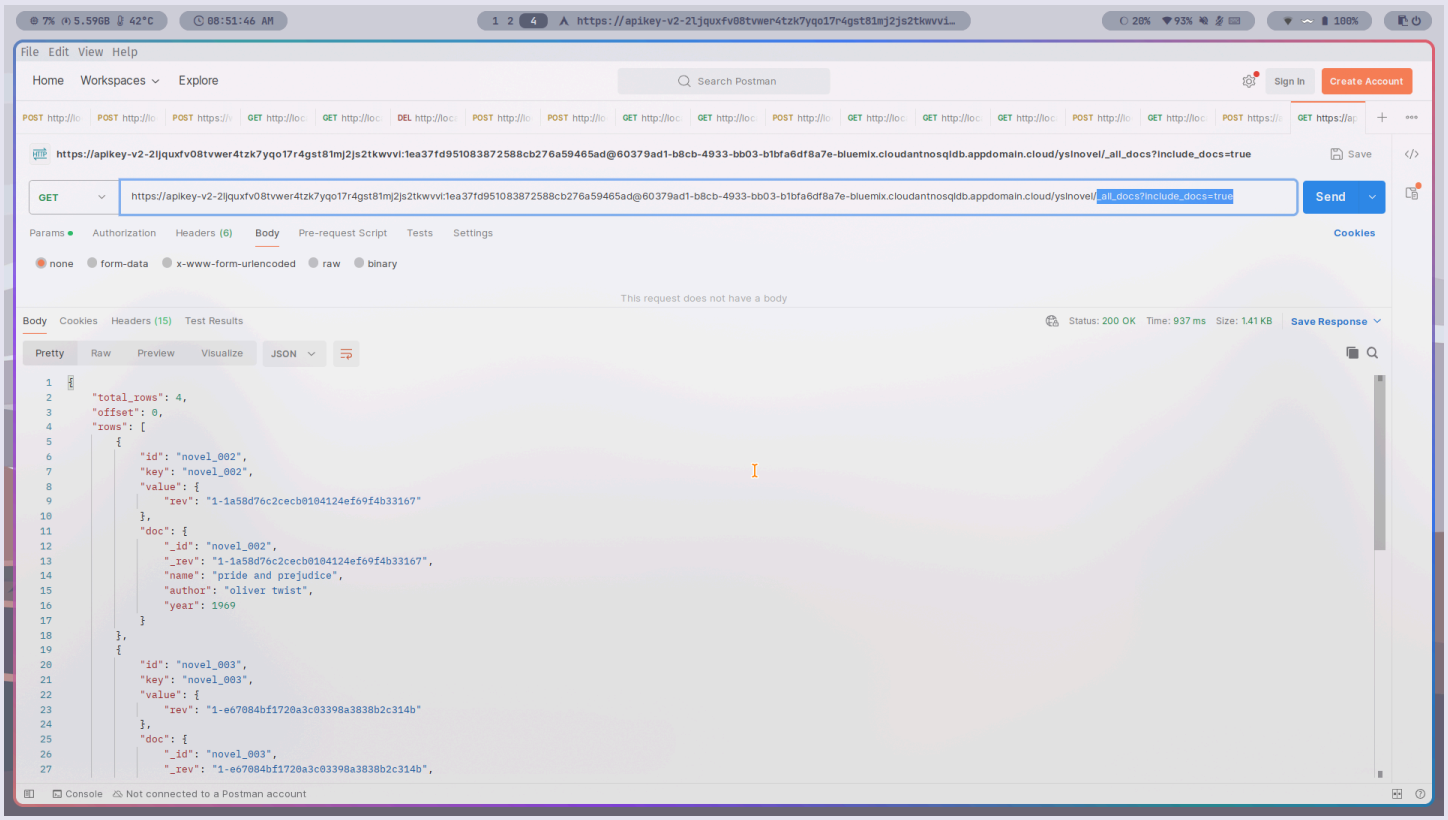
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

15. If GET request is sent to database/_all_docs then it returns the doc's details like ID and all without including the actual document



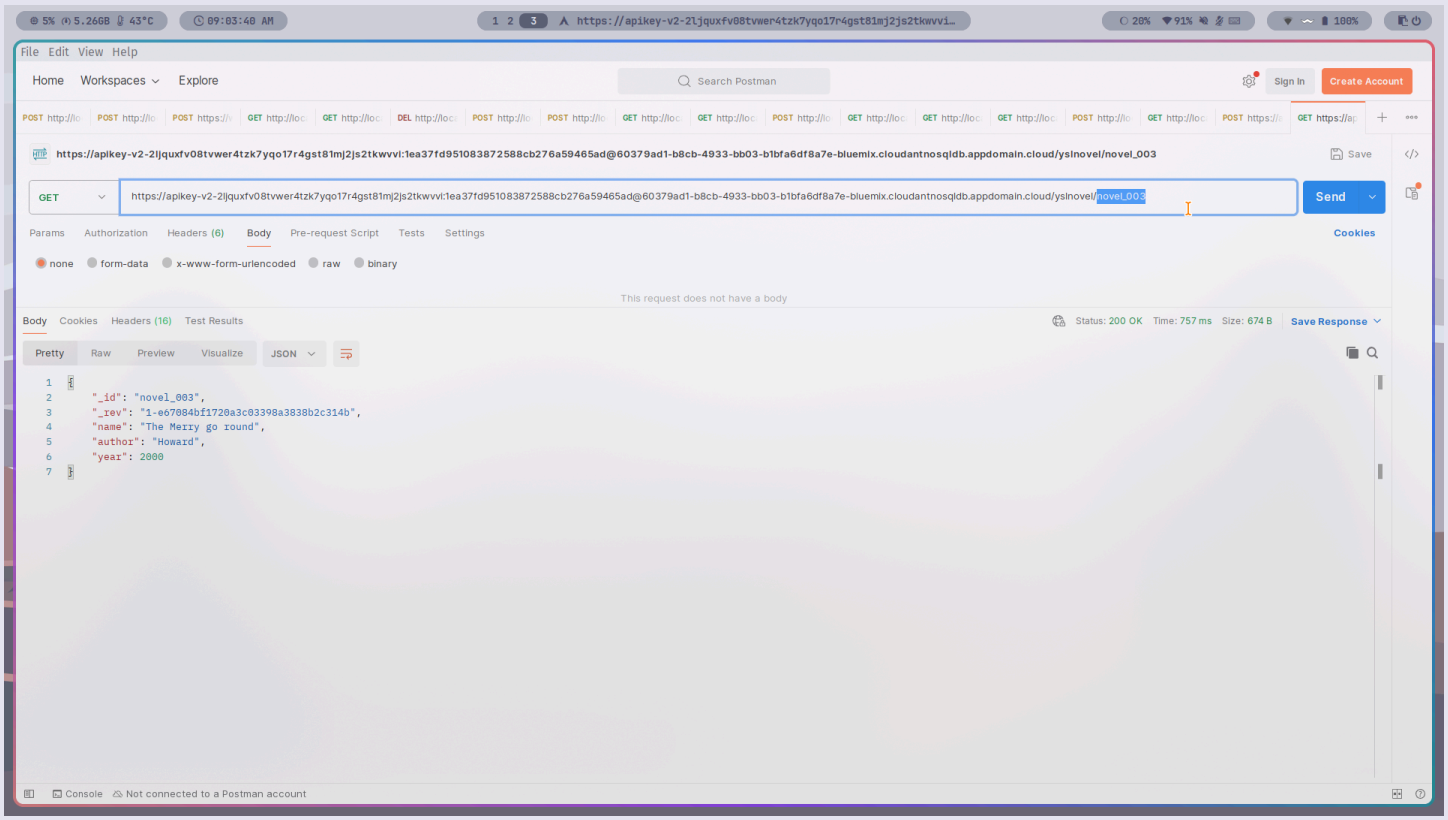
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

16. But, if include_docs=true is used as URN, then it also returns the actual document details



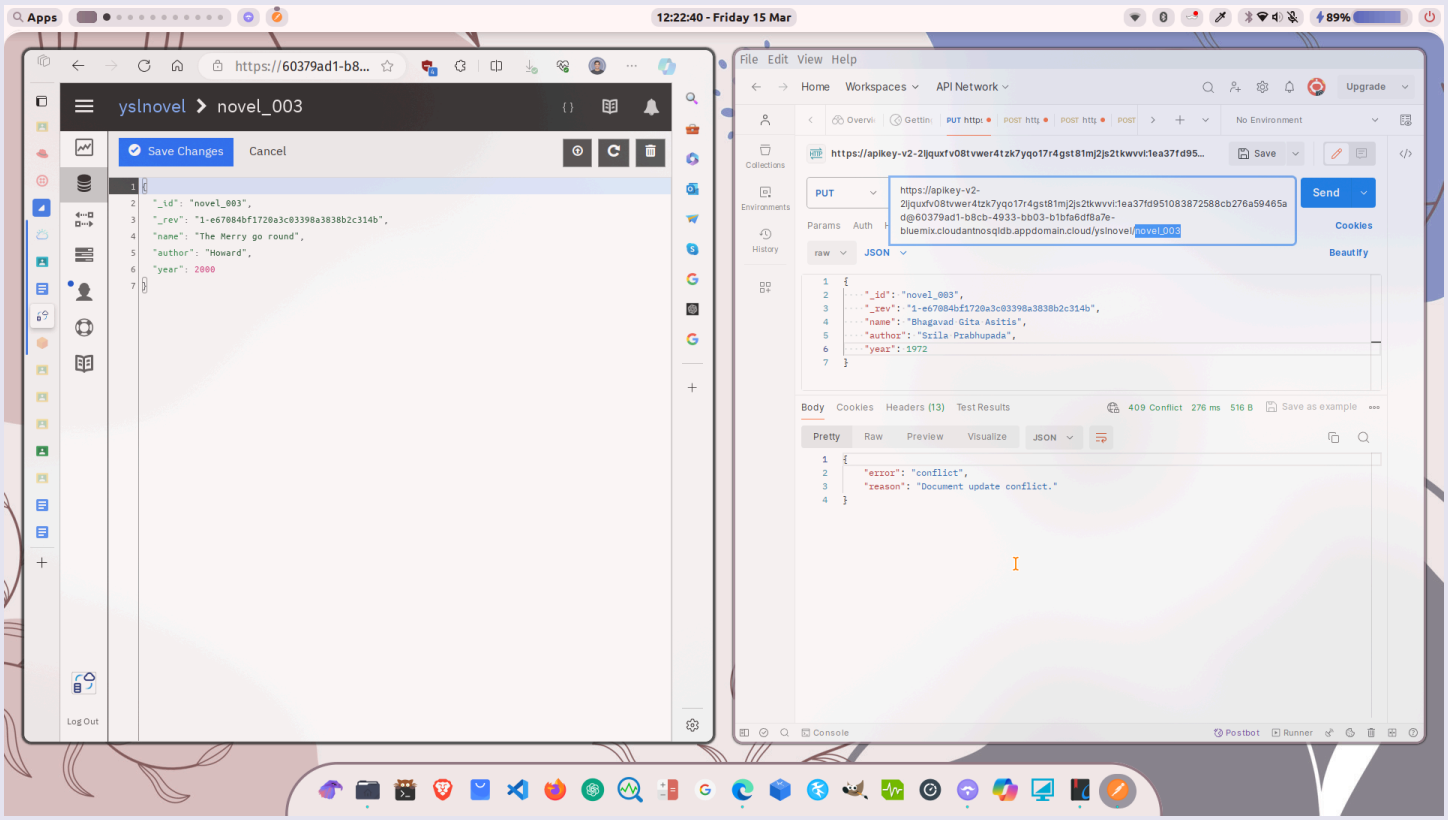
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

17. To get the specific document's details, enter its ID after URL/dbname/docID and make a GET request



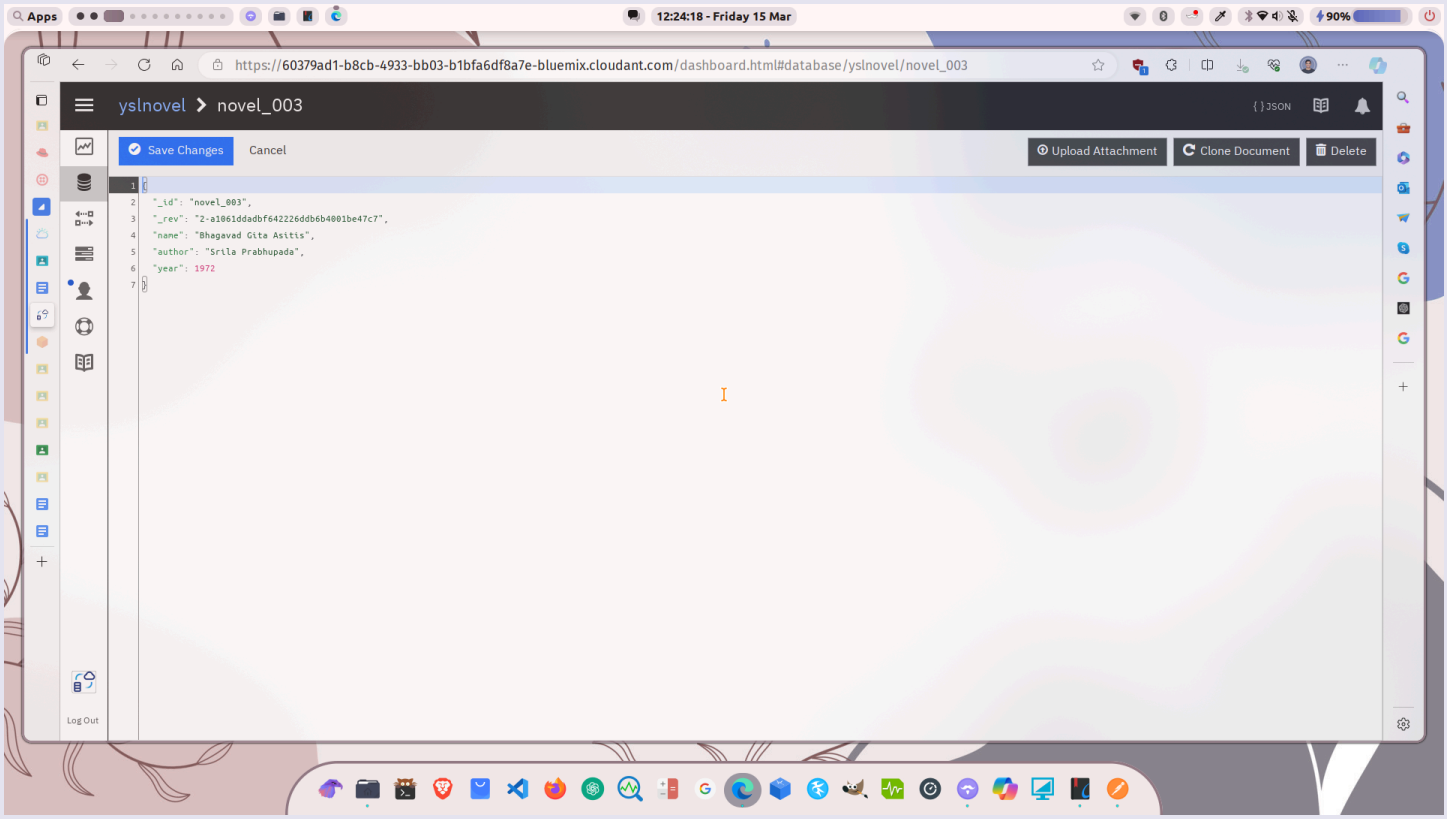
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

18. To update details, make a PUT request on same URL, but include previous revision ID in the json in body



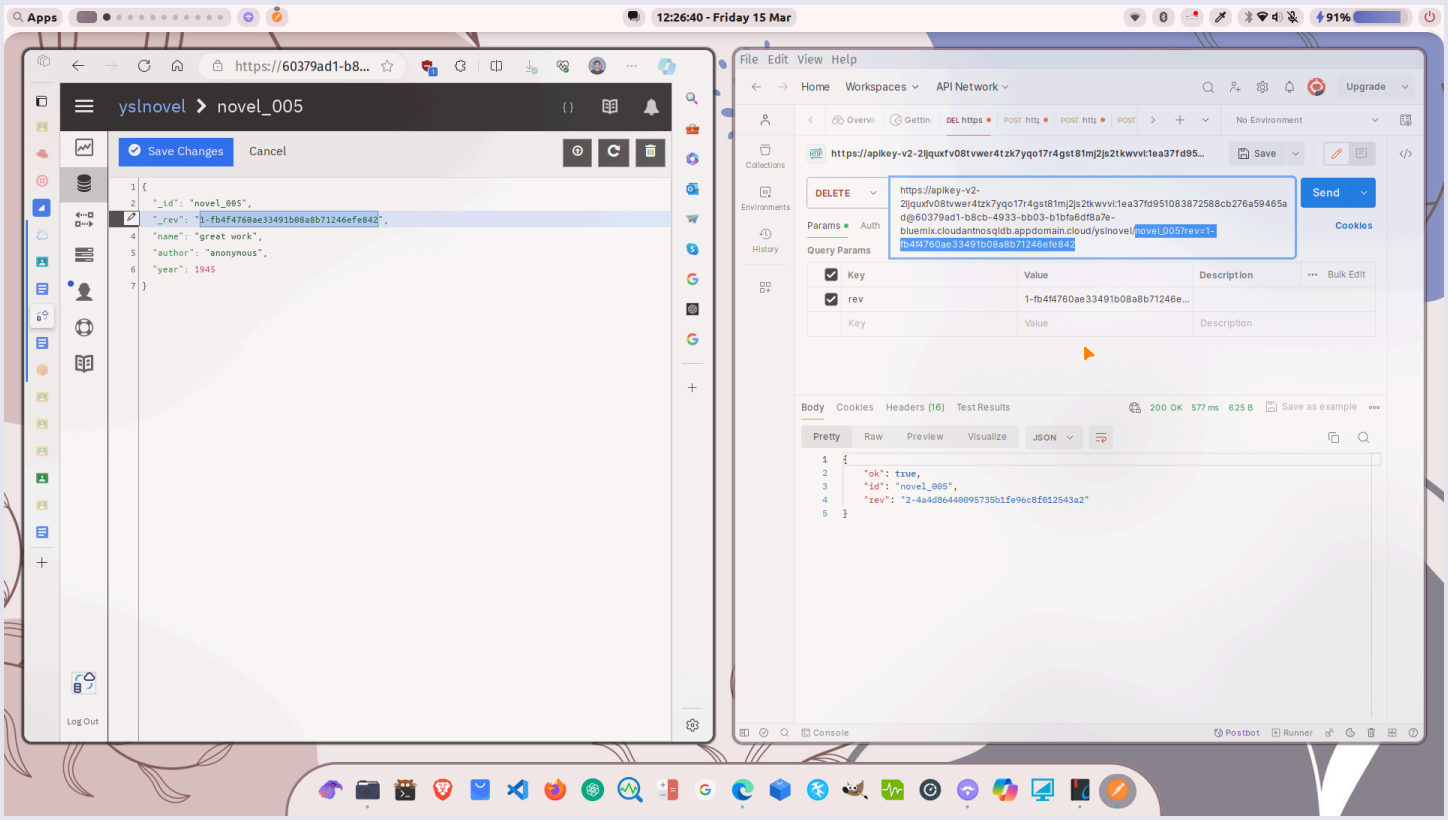
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

19. Now, the details will be updated in Cloudant database



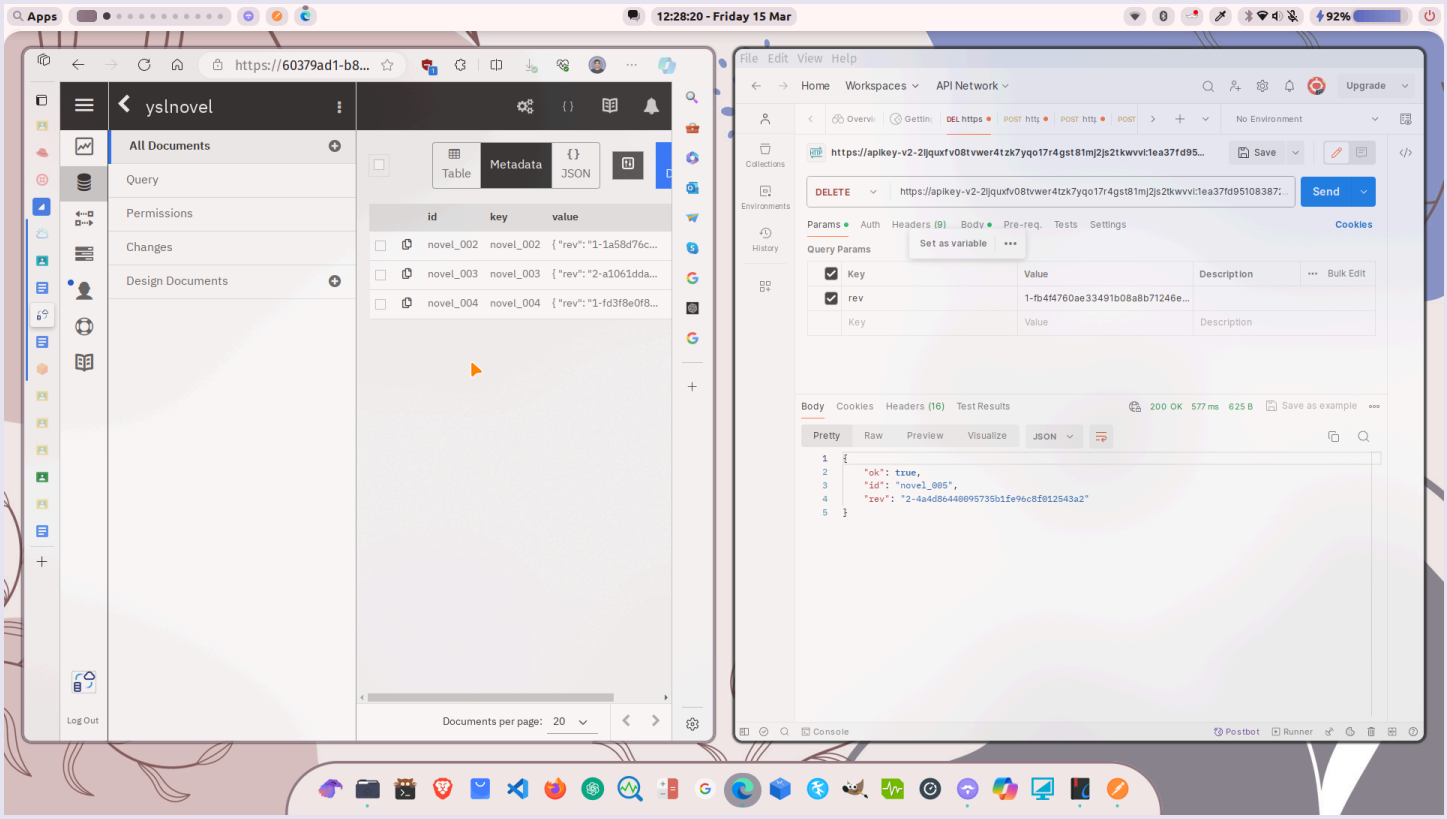
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

20. Now, to delete a record, same URL but with rev={revID} is required for DELETE request



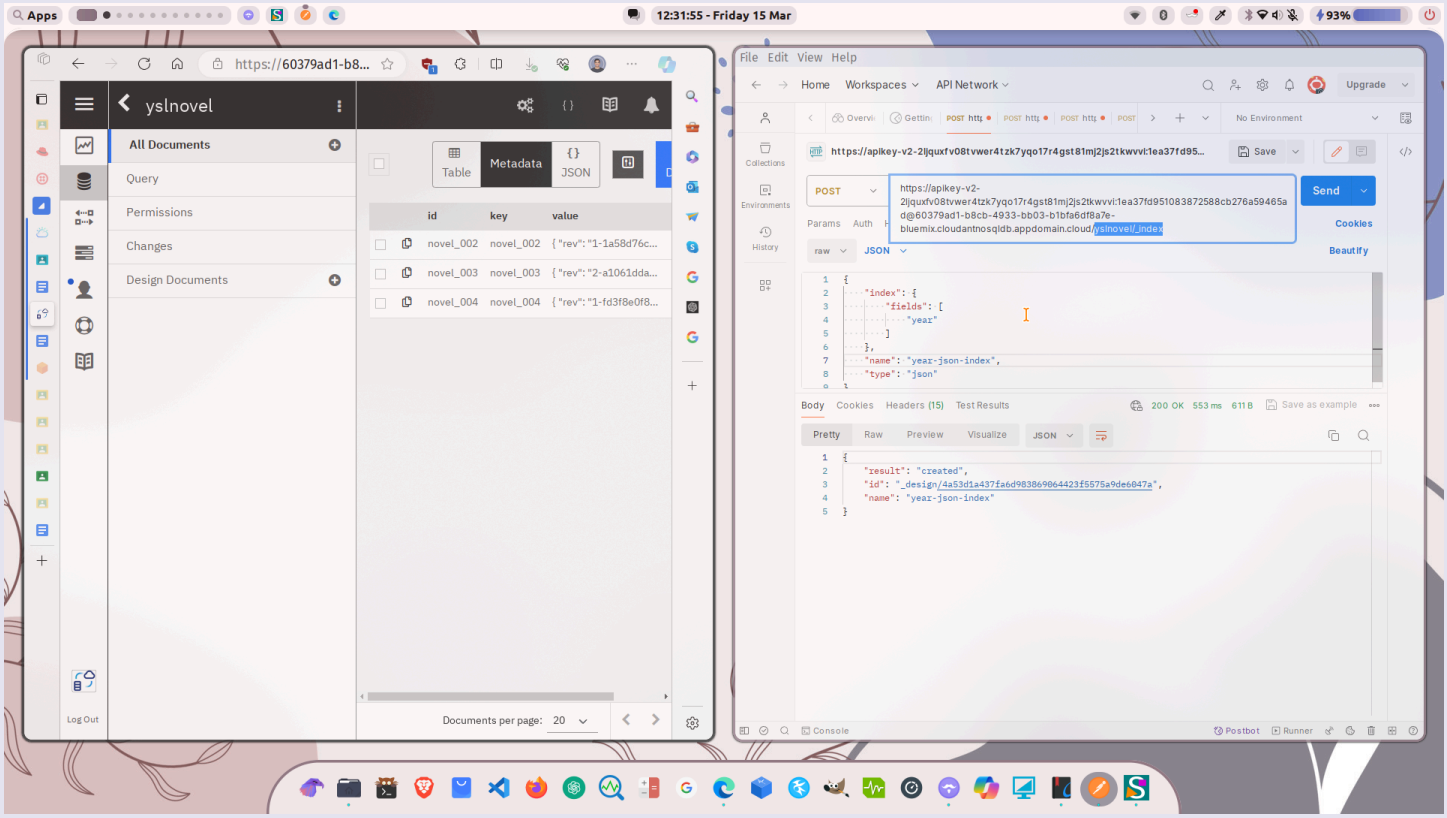
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

21. The document will be deleted now from the database



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

22. Here, for performing queries, we can create an index file for the same which includes the field(s) to perform query on, while the URI will be URL/dbname/_index and POST request should be made



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8

23. Now, for example if we want to filter data as per year, say year greater than 1864 and sort the data in ascending format, then the format of json body will below with URL/dbname/_find as URI with POST request :

```
{
    "selector": {
        "year": {
            "$gt": 1880
        }
    },
    "fields": [
        "_id",
        "name",
        "author",
        "year"
    ],
    "sort": [
        {
            "year": "asc"
        }
    ]
}
```

The index will also be visible on cloudant database

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 8