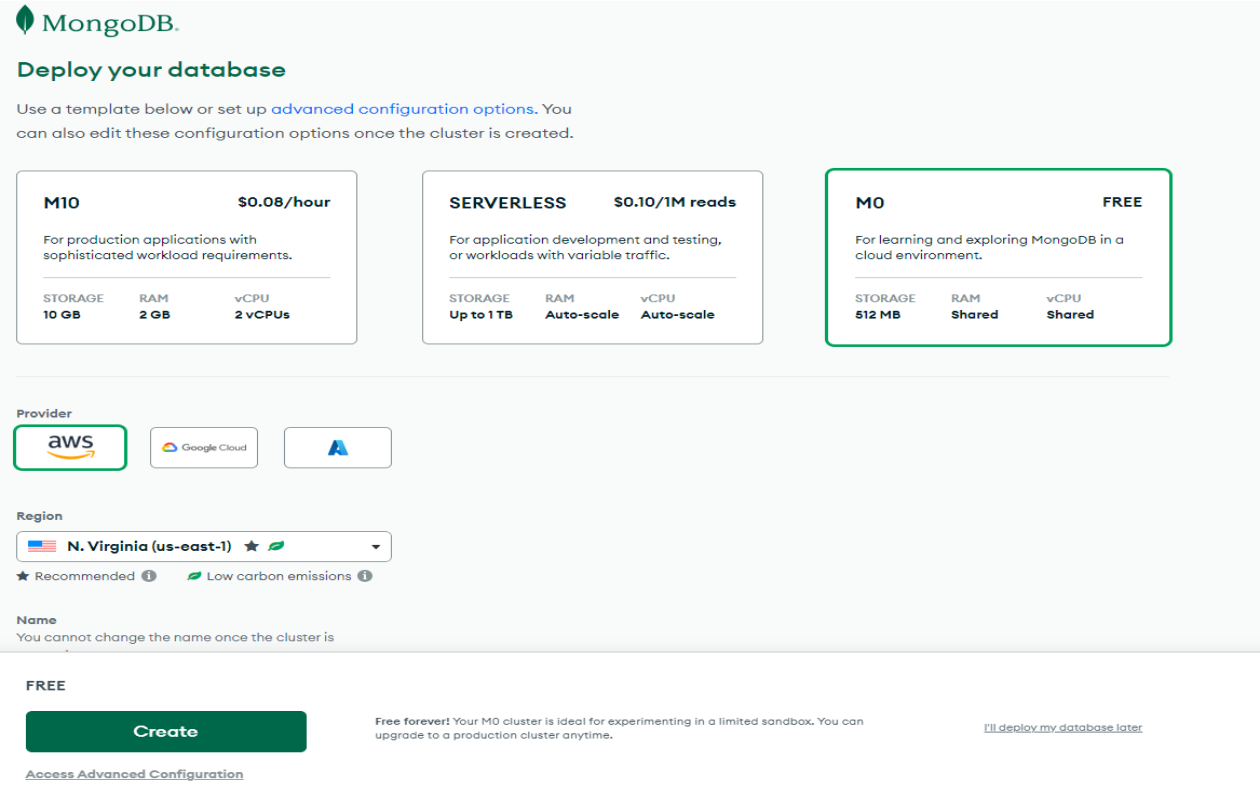
Steps to Connect to the Mongo DB

CONNECTIVITY TO MONGODB IN CLOUD:

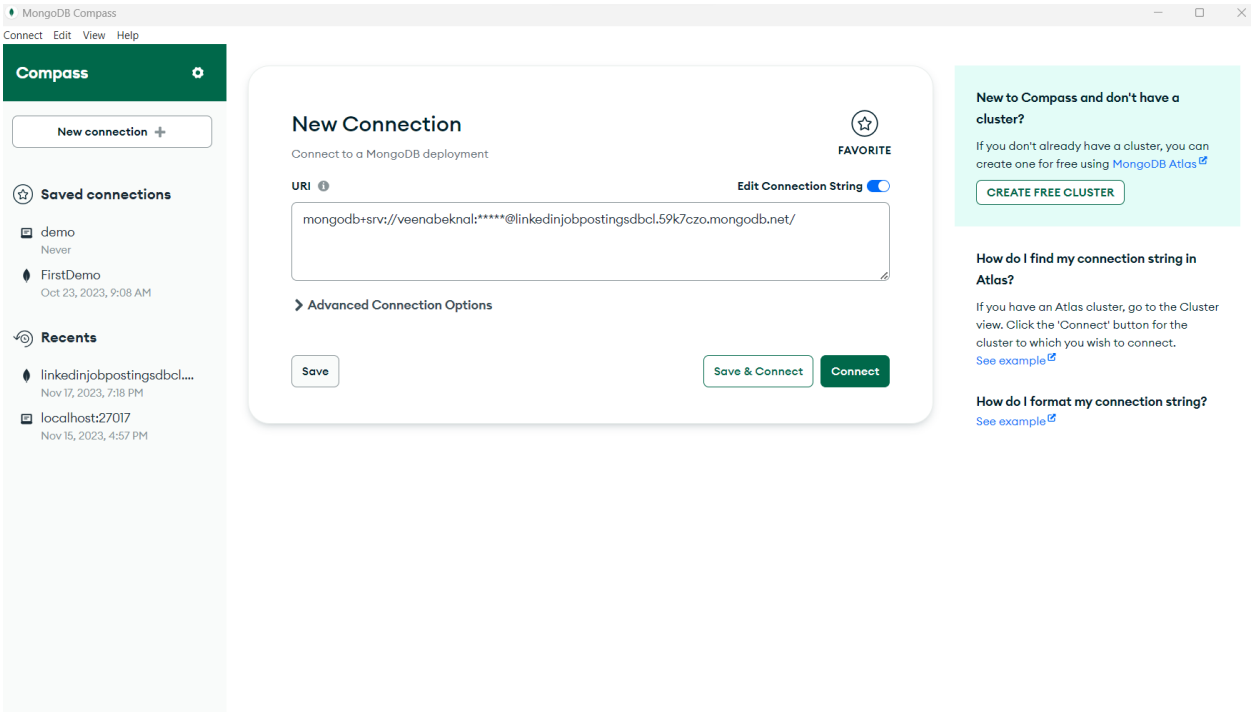
Post denormalization as explained in the above steps, we can create a cloud database instance containing the 3 collections. The cloud instance can be accessed through the Atlas interface (https://cloud.mongodb.com/). For this project, we are using the AWS provider's free tier as shown in the screenshot below:



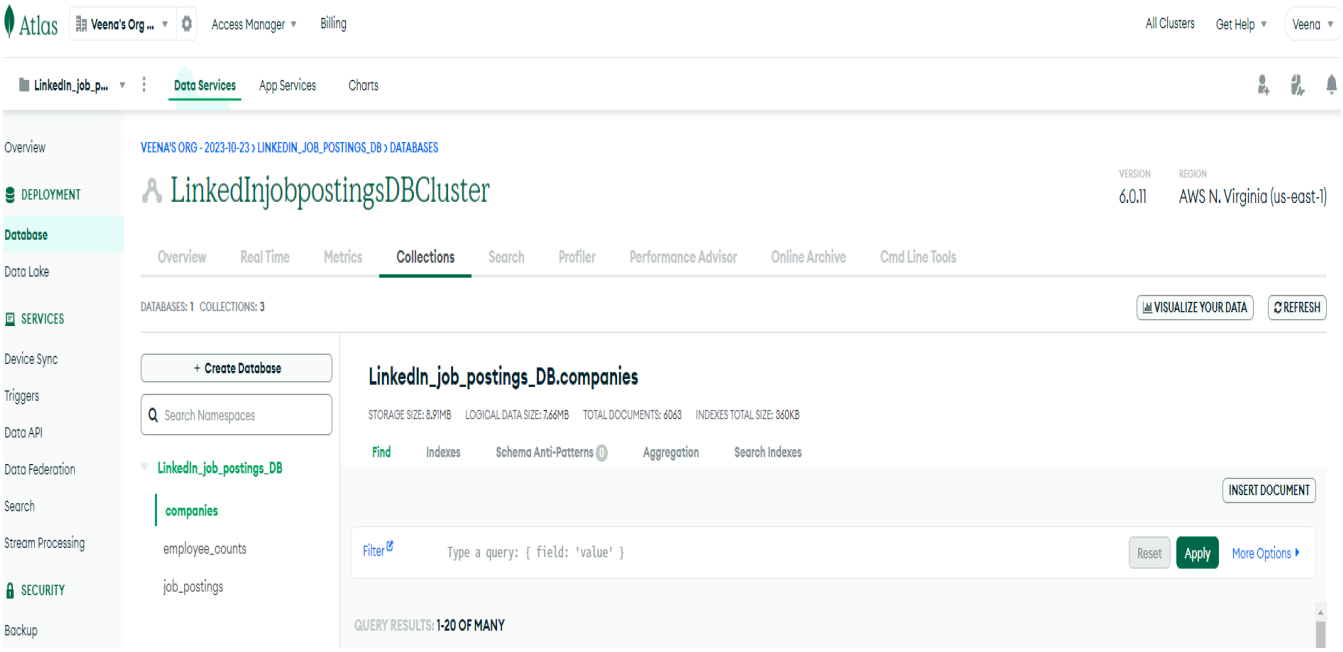
Once we have set up credentials, we can deploy a cluster with the required configuration. Our cluster named **LinkedInjobpostingsDBcluster** can be accessed using the following connection string:

mongodb+srv://veenabeknal:<password>@linkedinjobpostingsdbcl.59k7czo.mongodb.net/

Please note that the password has been masked for security reasons in the above connection string.



A snapshot of our cluster with the 3 collections is shown below:



There are 3 ways to input data into this cloud instance once the empty collections are created:

- Write out the 3 dataframes as CSV files and input them into the appropriate MongoDB collections using the Compass tool
- Same as step 1 except with JSON files

- Using the Pymongo python library, we can directly import our denormalized dataframes into the cloud instance using the connection string; the code used for job postings collection is shown below:

```
# Connection string for connecting to cloud instance of mongoDB using atlas
mongodb_client_atlas = MongoClient('mongodb+srv://veenabeknal:<password>@linkedinjobpostingsdbcl.59k7czo.mongodb.net/')

# Connecting the created database to cloud instance of mongoDB using atlas
db_ref_atlas = mongodb_client_atlas['LinkedIn_job_postings_DB']

# Connect to created empty collection to cloud instance of mongoDB using atlas
collection_job_atlas = db_ref_atlas['job_postings']

# Convert merged dataframe to a list of dictionaries
job_merge_dict = merged_job_postings_df.to_dict(orient="records")

# Iterate over each document to convert dictionaries to JSON arrays
# This is to ensure that the list fields type, industry_id and skill_abr are arrays
for doc in job_merge_dict:
    # Iterate over each field in the document
    for field, value in doc.items():
        # If the value is a dictionary, convert it to list
        if isinstance(value, dict):
            doc[field] = list(value.values())

    # Insert the document into the collection to mongoDB using atlas
    collection_job_atlas.insert_one(doc)
```

```
# Connect to created empty collection using the previously established connection to cloud instance of mongoDB
collection_companies_atlas = db_ref_atlas['companies']

# Convert merged dataframe to a list of dictionaries
companies_merge_dict = merged_companies_df.to_dict(orient="records")

# Iterate over each document to convert dictionaries to JSON arrays
# This is to ensure that the list fields specialities and industry are arrays
for doc in companies_merge_dict:
    # Iterate over each field in the document
    for field, value in doc.items():
        # If the value is a dictionary, convert it to list
        if isinstance(value, dict):
            doc[field] = list(value.values())

    # Insert the document into the collection to cloud instance of mongoDB
    collection_companies_atlas.insert_one(doc)
```

```
# Connect to created empty collection using the previously established connection to cloud instance of MongoDB atlas
collection_emp_count_atlas = db_ref_atlas['employee_counts']

# Convert merged dataframe to a list of dictionaries
emp_count_dict = employee_counts_df.to_dict(orient="records")

# Iterate over each document to convert dictionaries to JSON arrays
# This is to ensure that the list fields specialities and industry are arrays
for doc in emp_count_dict:
    # Iterate over each field in the document
    for field, value in doc.items():
        # If the value is a dictionary, convert it to list
        if isinstance(value, dict):
            doc[field] = list(value.values())

    # Insert the document into the collection to cloud instance of MongoDB atlas
    collection_emp_count_atlas.insert_one(doc)
```