NYC GenAI Hackathon Preparation Sheet

The purpose of this preparation sheet is to help participants get a lab environment set up and running before the event. We have provided some sample Python code that we can use as a template for some of the projects we'll be running in the hackathon. In addition to Llama 2, Bedrock, and others, we will try to provide alternative stacks at the event. Currently, the default stack consists of GPT3.5-turbo, Langchain, and MongoDB Atlas Vector Search.

NYC GenAI Hackathon Preparation Sheet Overview

Signup for a free **OpenAI** account
Create a Free **MongoDB Atlas** Account
Create or use an existing **Amazon** account
Create an **Amazon Sagemaker** Notebook Instance
Load and run the example **Jupyter Notebooks**

1) Signup for a free OpenAI account

- Create an account
- Select API keys in the left panel
- Verify by phone
- Create a secret key, save and copy it.

2) Create a Free MongoDB Atlas Account

Register a new account at: https://www.mongodb.com/cloud/atlas/register

Create an account

- Select Product from top down menu
- Select Try for Free Signup for a free account
- Great, now verify your email
- Fill in the little questionnaire
- Takes you into the Deploy Database

Deploy Database Cluster

- Select the free m0
- Use the defaultname *Cluster0*
- Default to N.Virginia (US-East-1)
- Keep *Automate security setup* enabled
- Keep *Add sample dataset* enabled
- Provide *AWS*
- Now create the deployment (button below/right)

Configure the database cluster

- There is an automatic user created.
    - (if copy and paste doesn't work, consider reloading the page)
- Fill out user and password & create the user
- Change the IP address that have access: (by clicking IP Access List)
    - by default it will fill in your current IP address
    - but for the hackathon we don't exactly what that IP will be
    - Therefore we allow all IPs
        - Use 0.0.0.0/0 for IP
        - do this for hackathon only , for production restrict this
- Now Create the Database and Cluster.

Get database cluster connection string

- After the Database is created select the *Connect* button for Cluster0
- Select Drivers
- Select Python (3.12 or later is fine)
- Copy the connect string: it Should look like this:

mongodb+srv://<username>*:<password>*@cluster0.faoasxh.mongodb. net/?retryWrites=true&w=majority

- Replace the <username> and <password> with the user you created

Congratulations, you're all done and the mongodb database setup is completed

Setup Atlas database

After the database Cluster is created and set up, we'll set up Atlas database:

- select Deployments > Database on the left side
- go over to the Brose *Collections* tab
- Select Add My Own Data
- *create database:*
- database ***langchain_db***
- collection_name ***test***

Setup Atlas search vector index

After the Database is created and set up, we'll set up an Atlas search vector index.

- select Atlas Search on the left navigation side
- select your database source: **Cluster0**
- Click on “Create Search Index”
- Select Atlas **Vector** Search **(Json editor)**
- Select Database ***langchain_db*** and collection_name ***test***
- IndexName ***index_name***
- Use the following json to configure:

```
{
 "fields": [{
 "numDimensions": 1536, "path": "embedding",
 "similarity": "cosine", "type": "vector"
 }]
}
```

Here’s what it should look like when you're done:

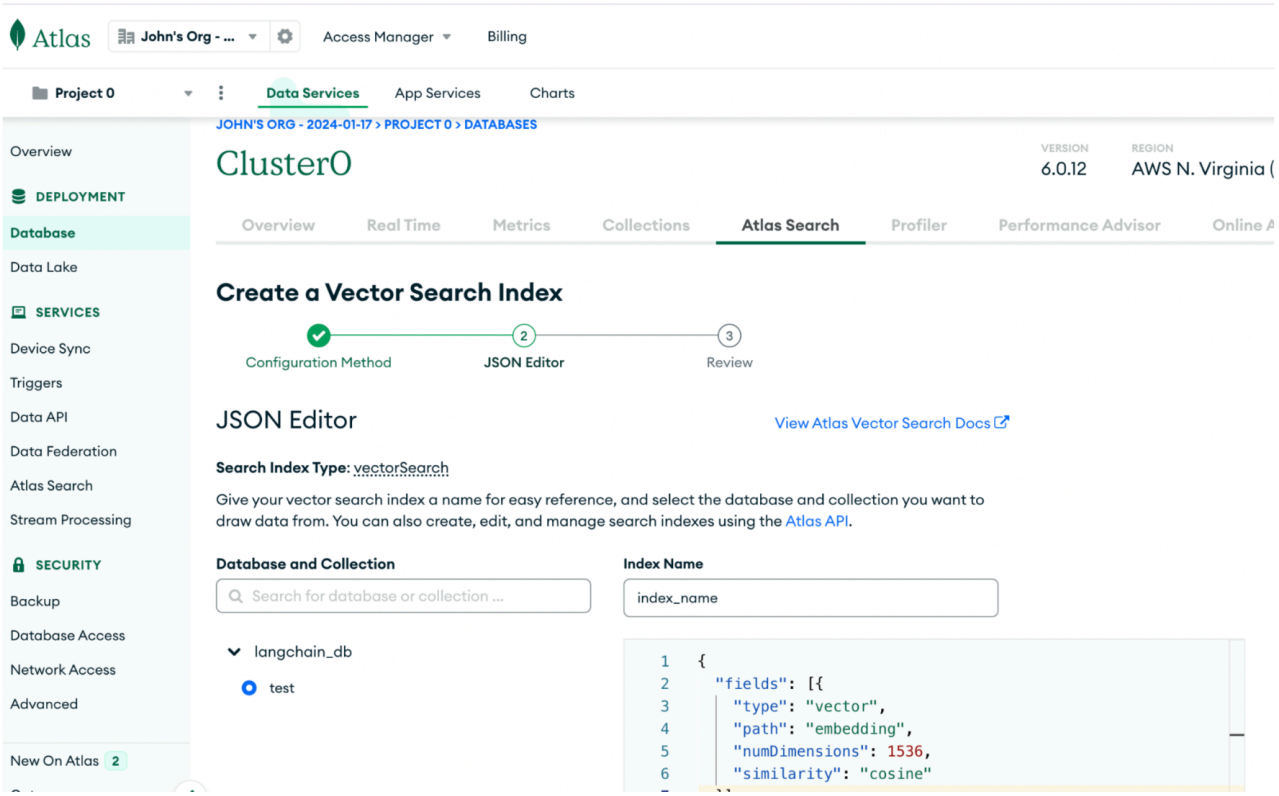

Figure 1

More reading

- For more details see https://www.mongodb.com/docs/atlas/atlas-vector-search/create-index/
- https://www.mongodb.com/blog/post/dedicated-search-nodes-vector-search-now-in-general-availability

3) Amazon Sagemaker

From the AWS Console navigate the Amazon Sagemaker (note you can use the search bar.)

Setup a Notebook Instance

After the database Cluster is created and set up, we'll set up a Notebook Instance:

- select Notebook > Notebook Instances on the left side
- Select the “Create notebook Instance” buttonin top right

- In Notebook Instance Settings
    - Add a Notebook Instance name (e.g., MDB-test1)
    - Use the defaults for the other setting fields
    - See Figure 2.
- In Permissions and encryption
    - Select “Create a new role” from dropdown (take the defaults)
    - See Figure 3

Configure a Git Repository

- *Select Git repository from the left window*
- *Select the Default repository arrow*
- *Select Clone a public Git repository to this notebook instance only*
- *Paste the repo*
    - *https://github.com/OperationalizingAI/Hackathon-1-24-24.git*
- *See Figure 4*

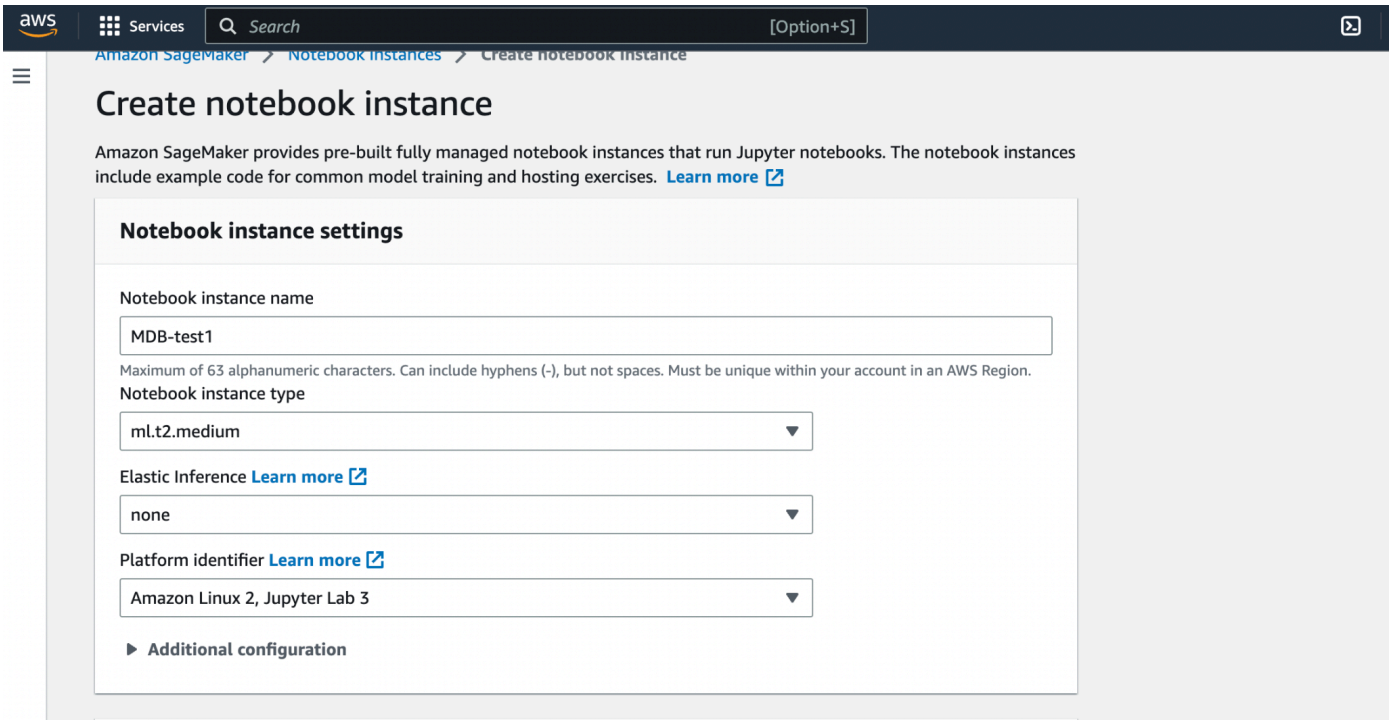

Figure 2

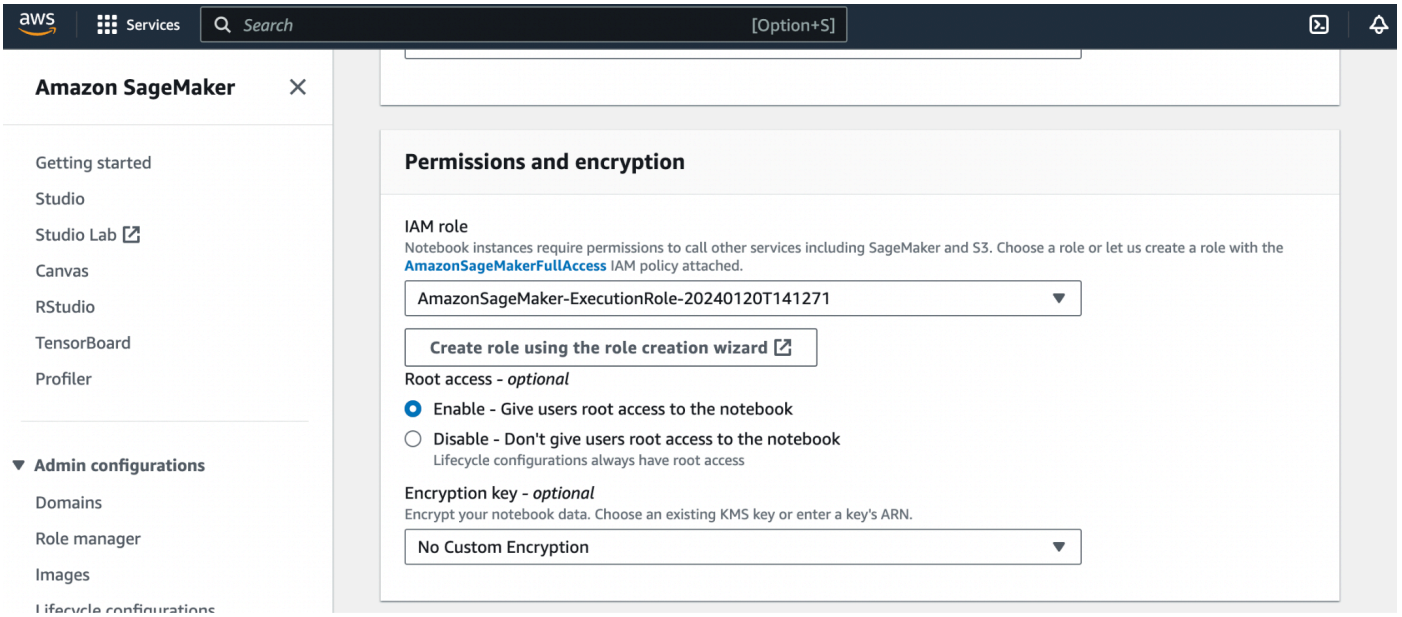

Figure 3

https://github.com/OperationalizingAI/Hackaton-1-24-24.git

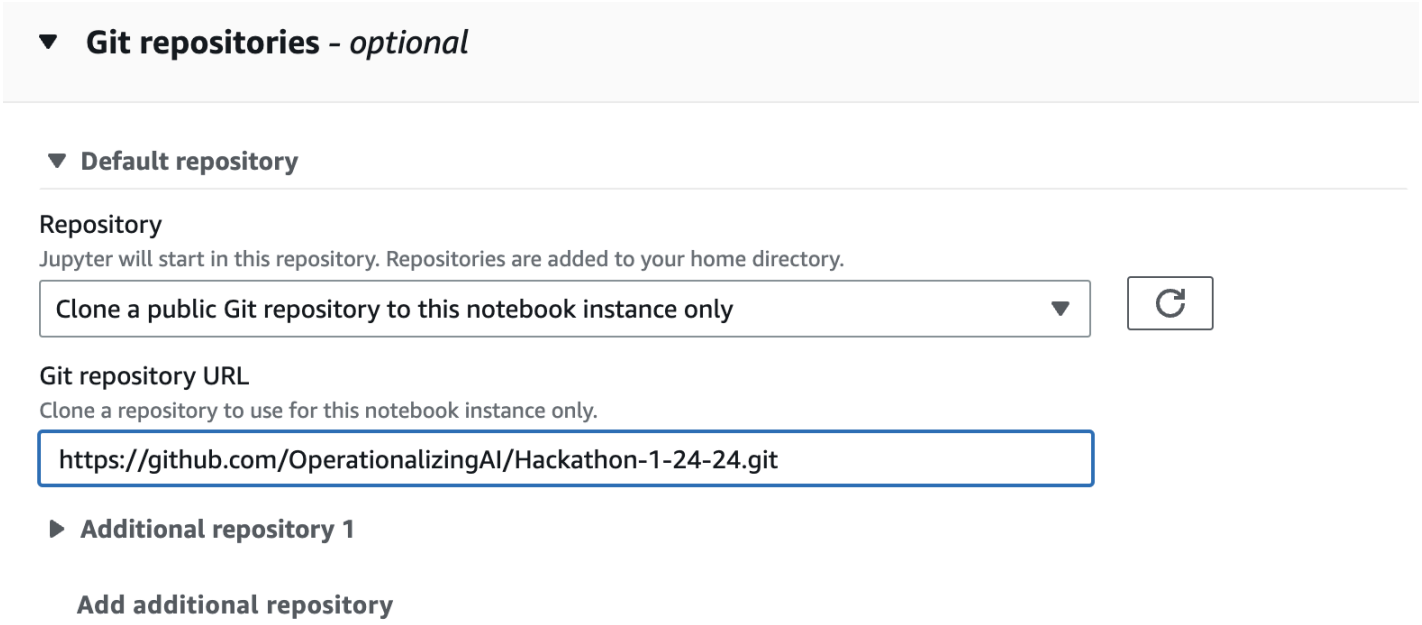

Figure 4

Create the Notebook

After the instance is running (inService) create the Python notebook.

- *Select the Open JupyterLabs action for the instance*
- *Under Notebook select the conda_python3 box*
- *Rename your notebook (e.g., MDB-test-1.ipynb)*
- *Import the sample notebook provided with the workshop*
- *Load the same note books for Colab*

        - *MDBLoad-SM.ipynb*
        - *DBRetrieve-SM.ipynb*
    - *Run your code.*

4) Google Colab (We are not using Colab for the NYC Hackathon)

Create a free account on Google Colab

- *Goto (https://colab.research.google.com/)*
- *Load the same note books for Colab*
    - *MDBLoad-GC.ipynb*
    - *DBRetrieve-GC.ipynb*
- *Run your code.*
-
- *Under Notebook select the conda_python3 box*
- *Rename your notebook (e.g., MDB-test-1.ipynb)*
- *Import the sample notebook provided with the workshop*
- *Run your code.*

5) Jupyter Notebooks

Example Code to load data in our database

https://colab.research.google.com/drive/18Qf2lZjV9FY6VdcDTDFdGk3NSNiAT2-3?usp=sharing

Example Code to query our data using RAG(Retrieval Augmented Generation)

https://colab.research.google.com/drive/10uYyKxYMgi1qkWSaq1WO8lA2X5SSPayw?usp =sharing

6) Addtional Notes for Processing Large Files

Setup a shared S3 blob

- *Bucket*
    - *Large-blobs*
    - *Upload the file(s)*
    - *Uncheck - Block all public access*
    - *Take all the default*
    - *Create Bucket*

Configure Bucket policy

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
```

```
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::large-blobs/*"
        }
    ]
}
```

From the EC2 Instance…
aws s3 cp s3://large-blobs/void.tar.gz ~/void.tar.gz