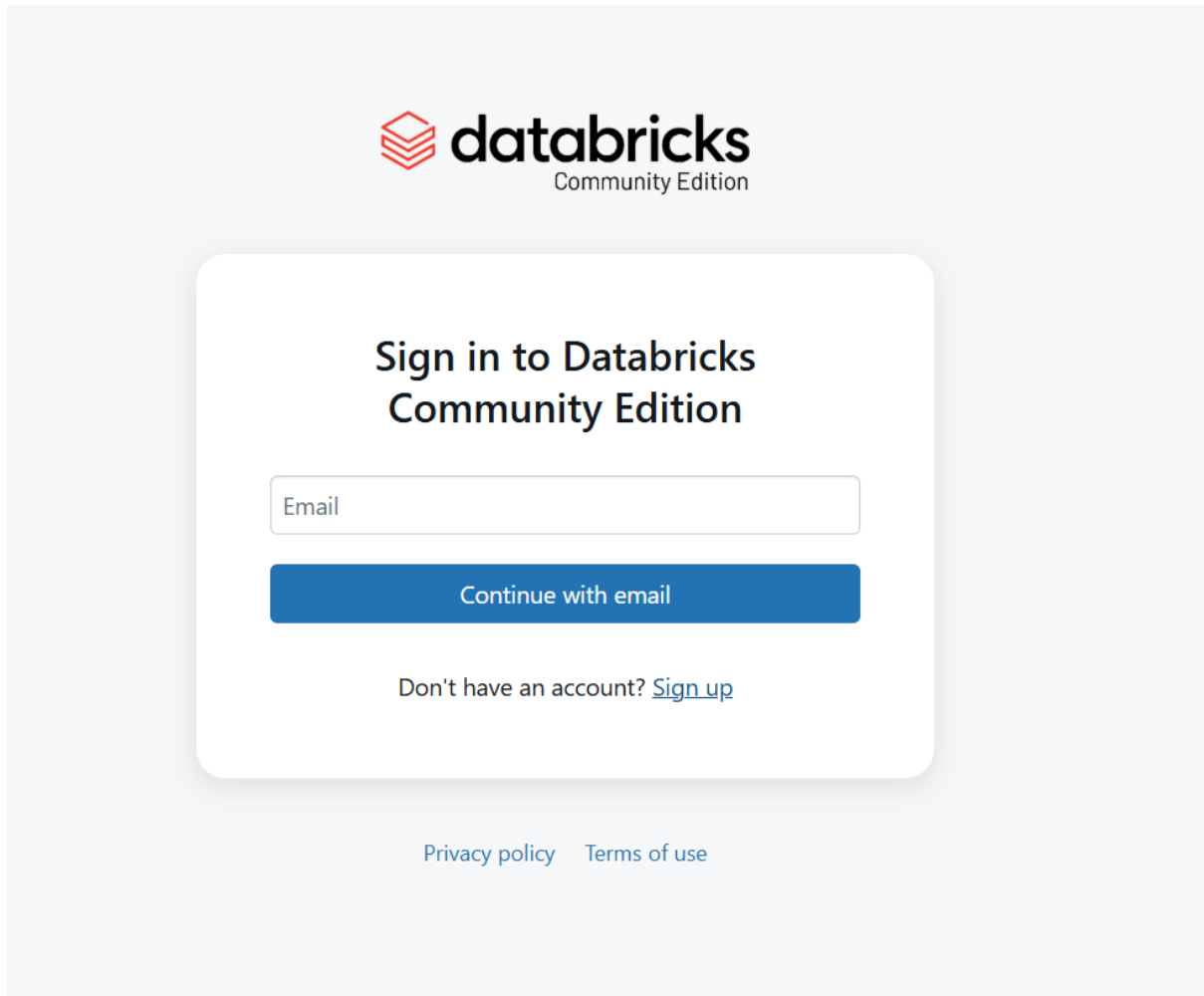


## Databricks and python

Create a Databrick community account:

Use link <https://community.cloud.databricks.com/>

Then go to link and click on don't have account

The image shows the login page for Databricks Community Edition. At the top, there is the Databricks logo (a red cube icon) followed by the text "databricks" in a bold, sans-serif font, and "Community Edition" in a smaller font below it. The main content is centered within a white rounded rectangle on a light gray background. It features the heading "Sign in to Databricks Community Edition" in a bold, dark blue font. Below this is a text input field with the placeholder "Email". Underneath the input field is a solid blue button with the white text "Continue with email". Below the button, there is a link that says "Don't have an account? [Sign up](#)". At the bottom of the page, there are two links: "Privacy policy" and "Terms of use", both in a small, blue font.

Signup using your gmail account



## Sign up for Community Edition

By clicking "Continue with email," you agree to Databricks' [Terms of Service](#) and [Privacy Notice](#).

Continue with email

[Go back](#)

Click on continue with email and will receive a code and use that code then will receive an email stating account is created then login using the email which we used for signup.



## Sign in to Databricks Community Edition

Continue with email

Don't have an account? [Sign up](#)

[Privacy policy](#) [Terms of use](#)

## Enter verification code

 - 

We sent an email with the code to  
**deepthi.kanamata2307@gmail.com**

Resend code in 28

Will receive the code and use it to login

Then create a cluster

Go to compute-> click in create compute to create a cluster

Compute

### deepthi.kanamata2307@gmail.com's Cluster •



Terminate

Edit

Configuration Notebooks (0) Libraries Event log Spark UI Driver logs Metrics Apps Spark compute UI - Master ▾

Databricks Runtime Version

12.2 LTS (includes Apache Spark 3.3.2, Scala 2.12)

Driver type

Community Optimized

15.3 GB Memory, 2 Cores

Instance

Free 15 GB Memory: As a Community Edition user, your compute will automatically terminate after an idle period of one or two hours.  
For more configuration options [↗](#), please [upgrade your Databricks subscription](#). [↗](#)

Spark JDBC/ODBC

Spark config ⓘ

spark.databricks.rocksDB.fileManager.useCommitService false

Environment variables ⓘ

PYSPARK\_PYTHON=/databricks/python3/bin/python3

Now create a notebook to work on python

Go to work book create new notebook

Number Datatypes in python

## DataTypes

▶

✓ 2 minutes ago (2s)

2

```
a=10
type(a)
```

Out[1]: int

▶

✓ 1 minute ago (<1s)

3

```
c= 10.5
type(c)
```

Out[3]: float

⋮

▶

✓ 1 minute ago (<1s)

4

Python

🗑️

🔗

⋮

```
d= 10+3j
type(d)
```

Out[5]: complex

Text data type in Python is String, which can be represented using single quotes, double quotes, triple single or double quotes

< > + Code + Text

▶

✓ 3 minutes ago (<1s)

5

```
b = "Deepthi"
type(b)
```

Out[2]: str

## Boolean Data type

▶

✓ 3 minutes ago (<1s)

5

```
b = "Deepthi"
type(b)
```

Out[2]: str

▶

✓ 2 minutes ago (<1s)

6

```
e= True
type(e)
```

Out[7]: bool

## Sequence Datatypes:

```
▶ ✓ 2 minutes ago (<1s) 7

fruits = ["apple", "Manago", "Banana"]
type(fruits)

Out[8]: list
```

```
⋮ ▶ ✓ Just now (<1s) 8 Python 🗑️ 📄 ⋮

names= ("Deepthi", "Rahul", "Ann")
type(names)

Out[10]: tuple
```

## Dictionary Datatype:

```
⋮ ▶ ✓ Just now (<1s) 9 Python 🗑️ 📄 ⋮

students={"name": "Deepthi", "ID": 10}
type(students)

Out[11]: dict
```

## Identifiers:

### Rules for Python variables:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_)
- Variable names are case-sensitive (age, Age and AGE are three different variables)
- A variable name cannot be any of the [Python keywords](#).

### Python key words are:

```
⋮ ▶ ✓ Just now (<1s) 11 Python 🗑️ 📄 ⋮

import keyword
keywords = keyword.kwlist
print("Python Keywords:")
print(keywords)

Python Keywords:
['False', 'None', 'True', '__peg_parser__', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```