

AMI Name: Gigabits-DeepSeek-Ubuntu 24.04

1. Lunch Instance with above AMI

Default Username: ubuntu

Minimum Size: 20GiB (more prefer depend on module)

Recommended RAM: 32GiB (Recommended more)

Recommended CPU: Higher as much as possible (GPU Recommended)

2. Verification:

i. **systemctl status ollama**

```
root@ip-172-31-26-199:~# systemctl status ollama
● ollama.service - ollama service
   Loaded: loaded (/etc/systemd/system/ollama.service; enabled; preset: enabled)
   Active: active (running) since wed 2025-01-29 08:02:19 UTC; 44s ago
     Main PID: 617 (ollama)
       Tasks: 8 (limit: 38494)
      Memory: 33.9M (peak: 34.0M)
         CPU: 49ms
       CGroup: /system.slice/ollama.service
              └─617 /usr/local/bin/ollama serve

Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: [GIN-debug] GET    /api/tags                --> github.com/ollama/ollama/server.(*Server).ListHan
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: [GIN-debug] GET    /api/version             --> github.com/ollama/ollama/server.(*Server).Generat
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: [GIN-debug] HEAD    /                        --> github.com/ollama/ollama/server.(*Server).Generat
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: [GIN-debug] HEAD    /api/tags                --> github.com/ollama/ollama/server.(*Server).ListHan
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: [GIN-debug] HEAD    /api/version             --> github.com/ollama/ollama/server.(*Server).Generat
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: time=2025-01-29T08:02:20.1242 level=INFO source=routes.go:1238 msg="Listening on 127.0.0.1:11434 (S
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: time=2025-01-29T08:02:20.1282 level=INFO source=routes.go:1267 msg="dynamic LLM libraries" runners
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: time=2025-01-29T08:02:20.1322 level=INFO source=gpu.go:226 msg="looking for compatible GPUs"
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: time=2025-01-29T08:02:20.1552 level=INFO source=gpu.go:392 msg="no compatible GPUs were discovered"
Jan 29 08:02:20 ip-172-31-26-199 ollama[617]: time=2025-01-29T08:02:20.1552 level=INFO source=types.go:131 msg="inference compute" id=0 library=
lines 1=20/20 (END)
```

ii. **ollama list** (to check the available module)

```
root@ip-172-31-26-199:~# ollama list
NAME                ID                SIZE      MODIFIED
deepseek-r1:1.5b    a42b25d8c10a     1.1 GB    9 minutes ago
deepseek-r1:7b      0a8c26691023     4.7 GB    32 minutes ago
root@ip-172-31-26-199:~# |
```

iii. **ollama pull** (module name)

E.g. **ollama pull deepseek-r1:671b**

[Link to Deepseek-R1](#)

The size depends on the module you want to use. The default modules available on the system are **1.5B** and **7B**. (You can verify this by running *ollama list*.)

iv. **ollama run** (your module name)

E.g. **ollama run deepseek-r1:7b**

```

root@ip-172-31-26-199:~# ollama run deepseek-r1:7b
>>> hi
<think>
</think>
Hello! How can I assist you today? 😊

>>> provide me python code, which will take 2 input from user and provides the sum in answer.
<think>
Alright, the user is asking for a Python program that takes two inputs from the user and provides their sum as an answer. Let's break down how to approach this.

First, I need to write code that can accept two different types of inputs. The user could input numbers directly, or they might provide strings which represent numbers. So, using `input()` functions makes sense here.

Next, converting these string inputs into integers is crucial because we want to perform a numerical addition. The `int()` function will handle the conversion seamlessly.

Then, adding the two integers together and storing the result in a variable named `sum` seems straightforward.

Finally, printing out the result using `print("Sum is:", sum)` will provide the desired output. I should make sure the message is clear and friendly to enhance user experience.

I should also consider potential edge cases, like non-integer inputs. However, since the user specifically mentioned taking two inputs and providing the sum, handling only numeric conversions might be sufficient for now, unless specified otherwise.

Putting it all together, the code will prompt the user twice for input, convert those to integers, add them up, and display the result. This should meet the user's requirements effectively.
</think>

Sure! Here's a Python program that takes two inputs from the user and returns their sum:

```python
Get two numbers from the user
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

Calculate the sum
sum = num1 + num2

Print the result
print("Sum is:", sum)
```

This program will:
1. Ask the user to enter the first number and store it in `num1`.
2. Ask the user to enter the second number and store it in `num2`.
3. Add `num1` and `num2` together.
4. Print the result of the addition.

For example, if you run this program and input 5 and then 7, it will output:

```
Sum is: 12
```

>>> Send a message (/? for help)

```

v. **/bye** or **ctrl +d** for to exit from ai shell.

```

>>> /bye
root@ip-172-31-26-199:~# |

```

Thank you!