





### ? Resources:

- This section shows the **current status of system resources allocated** to your Google Colab session, including RAM and disk space. It helps you monitor the amount of computational power your notebook is using, which is especially important when working on tasks that consume a lot of memory or disk space.

### ? RAM and Disk Information:

- **System RAM:** Displays the current usage of Random Access Memory (RAM) out of the total available. For example, in your screenshot, it shows 1.1 / 12.7 GB, meaning that **1.1 GB of RAM is being used out of a total of 12.7 GB.**
- **Disk:** Displays the current disk space usage and the total available disk space. In your screenshot, it shows 33.1 / 107.7 GB, meaning **33.1 GB of disk space is being used out of 107.7 GB.**

### ? Backend Details:

- **Python 3 Google Compute Engine backend:** Indicates that the notebook is running on a backend provided by Google Compute Engine, which is a virtual machine optimized for running Colab notebooks.

## Google Colab Framework Overview

Google Colab, short for "Collaboratory," **is a cloud-based Jupyter notebook environment primarily used for running Python code.** It is part of the Google Research project and **provides free access to computing resources, including GPUs and TPUs,** making it a popular tool for **data analysis, machine learning, and educational purposes.** Below is a detailed look at the components and functionalities of Google Colab.

### Key Components of Google Colab Framework

#### 1. Notebook Interface:

- Similar to Jupyter Notebooks, Colab provides a cell-based interface where code, text, images, and other media can be combined.
- Cells can be of different types:
  - **Code Cells:** Used for writing and executing code.
  - **Markdown Cells:** Used for writing formatted text, including descriptions, equations, and links.

#### 2. Runtime Environment:

- **Colab provides a cloud-based runtime,** allowing users to execute code without needing local installations.
- The runtime can be customized by selecting different hardware accelerators:
  - **CPU:** Standard processing unit, suitable for basic tasks.
  - **GPU:** Graphics Processing Unit, ideal for parallel processing tasks, such as deep learning.
  - **TPU:** Tensor Processing Unit, specialized hardware for accelerating machine learning models, especially TensorFlow.

#### 3. Resource Management Panel:

- **Located on the right side of the notebook,** **this panel provides insights into the current usage of resources:**

- **RAM Usage:** Displays the current memory utilization.
  - **Disk Usage:** Shows the current disk space used by the session.
  - **Session Duration:** Indicates the runtime's expected duration based on the current usage and the tier (free or Colab Pro).
  - **Provides options to manage runtime:**
    - **Restart Runtime:** Resets the environment, clearing all variables and reloading the installed packages.
    - **Change Runtime Type:** Allows switching between CPU, GPU, and TPU.
4. **Package Management:**
- Colab supports the installation of external libraries using package managers like pip and apt-get.
  - Each session starts fresh, so packages need to be reinstalled in each new session unless saved through persistent storage.
5. **Persistent Storage:**
- Colab's environment resets when the session ends, so it provides options to connect with persistent storage:
    - **Google Drive Integration:** Easily mount Google Drive to save or load files directly from your drive.
    - **Local Files:** Upload or download files directly to/from your notebook.
6. **Code Execution and Visualization:**
- Supports various programming languages such as Python, Java, R, and others with the proper setup.
  - Provides rich output support, including charts, tables, images, and interactive visualizations through libraries like Matplotlib, Seaborn, and Plotly.
7. **Collaboration Features:**
- Multiple users can work on the same notebook simultaneously, making it easy to share and collaborate.
  - Integration with Google services (Drive, Docs) makes it seamless to share work with others.

-----

# Step 1: Install Java (OpenJDK 17)

```
!apt-get install openjdk-17-jdk-headless -qq > /dev/null
```

# Step 2: Write the Java code to a file (HelloWorld.java)

```
%%writefile HelloWorld.java
```

```
public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello, World from Google Colab!");

    }

}
```

# Step 3: Compile the Java program

```
!javac HelloWorld.java
```

# Step 4: Run the Java program

```
!java HelloWorld
```

#### Explanation

1. **Installation:** `!apt-get install openjdk-17-jdk-headless -qq > /dev/null` installs Java 17 without displaying output details (`-qq > /dev/null` makes the output quiet).
2. **Writing Java Code:**
  - `%%writefile HelloWorld.java` saves the Java code to a file named `HelloWorld.java`.

- The code defines a simple Java class with a main method that prints a message.
3. **Compiling:**
- `javac HelloWorld.java` compiles the Java file into bytecode, generating a `HelloWorld.class` file.
4. **Running:**
- `java HelloWorld` executes the compiled program, displaying the output.
- 

## Difference Between EC2 Instance and Google Colab

1. **Purpose and Use Case:**
  - **EC2 Instance (Amazon Elastic Compute Cloud):** EC2 is a service provided by AWS that allows users to run virtual servers in the cloud. It's designed for a wide range of applications, including hosting web applications, machine learning workloads, data processing, and running databases. EC2 gives users full control over the OS, software stack, and hardware configurations.
  - **Google Colab:** Google Colab is a free, cloud-based Jupyter notebook environment specifically designed for data science, machine learning, and AI research. It provides access to GPUs and TPUs, making it suitable for prototyping ML models and running Python scripts interactively.
2. **Control and Customization:**
  - **EC2 Instance:** Offers extensive control over the environment, including OS selection (Linux, Windows), custom hardware configurations (CPU, memory, storage), and software setup. Users can manage security, networking, and have root access to the system.
  - **Google Colab:** Limited control over the environment. Users can install Python packages but cannot change the underlying OS or significantly alter hardware configurations. The environment resets when idle for too long or when the session ends.
3. **Pricing:**
  - **EC2 Instance:** Pricing is based on the instance type (CPU, GPU, memory), usage duration, and other factors like storage and bandwidth. It's generally a pay-as-you-go model with the option to reserve instances for a lower cost.
  - **Google Colab:** Offers a free tier with limited resources (RAM, GPU availability). For more consistent access to better GPUs and longer session durations, users can subscribe to Colab Pro or Pro+, which come with monthly fees.
4. **Hardware and Performance:**
  - **EC2 Instance:** Wide variety of instance types with specific hardware configurations, including high-performance GPUs, CPUs, and large memory options tailored to compute-intensive tasks.
  - **Google Colab:** Provides access to a limited set of GPUs (NVIDIA K80, T4, P100, V100) and TPUs. Performance may vary, and resources are shared, which means that availability can be inconsistent, especially on the free tier.
5. **Persistence and Storage:**
  - **EC2 Instance:** Data and configurations are persistent as long as the instance is running, and storage can be configured using Elastic Block Store (EBS) volumes or other AWS storage solutions.
  - **Google Colab:** Does not persist files or data automatically between sessions unless connected to external storage solutions like Google Drive. Each session starts fresh, making it ideal for short-term projects and prototyping.
6. **Security and Access:**
  - **EC2 Instance:** Offers robust security features, including Virtual Private Clouds (VPCs), security groups, IAM roles, and encryption options. Users have fine-grained control over who can access the instance.
  - **Google Colab:** Simplified security model mainly involving Google Account authentication and permissions for accessing external storage like Google Drive.

## Summary

- **EC2 Instance:** Best suited for production workloads, applications requiring full environment control, and scalable cloud infrastructure.
- **Google Colab:** Ideal for interactive, experimental work in machine learning and data science, especially for students and researchers needing quick access to powerful hardware with minimal setup.

Yes, you will need to run the installation command (`!apt-get install openjdk-17-jdk-headless -qq > /dev/null`) each time you start a new Google Colab session. This is because Google Colab provides a fresh, temporary environment every time you open a new notebook or restart the runtime. When the session ends, all installed software and files that are not explicitly saved to external storage (like Google Drive) are lost.

Console Home | Console | EC2 Instance Connect | ChatGPT | (43) WhatsApp | Welcome To Colab - Colab | Untitled0.ipynb - Colab

eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-06380ffc1507e4425&osUser=ec2-user&region=eu-north-1&sshPort=22#

Services Search [Alt+S]

Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

Last login: Thu Sep 12 14:04:30 2024 from 13.48.4.202  
(ec2-user@ip-172-31-34-170 ~) \$ cal 09 2024  
September 2024  
Su Mo Tu We Th Fr Sa  
1 2 3 4 5 6 7  
8 9 10 11 12 13 14  
15 16 17 18 19 20 21  
22 23 24 25 26 27 28  
29 30

(ec2-user@ip-172-31-34-170 ~) \$ date  
Thu Sep 12 15:10:07 UTC 2024  
(ec2-user@ip-172-31-34-170 ~) \$

i-06380ffc1507e4425 (barshaserver)  
PublicIP: 13.60.77.168 PrivateIP: 172.31.34.170

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG INTL 20:40 12-09-2024