

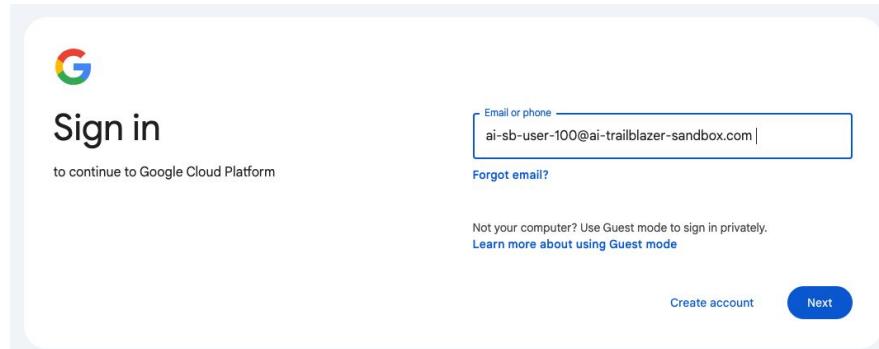
LAB GUIDE

AI Labs - Prudential with Google

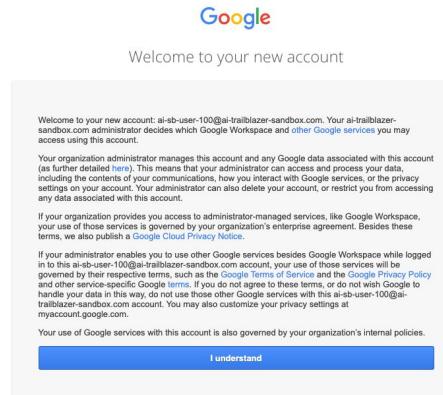
Labs Access: Accessing Google Cloud Console

Step 1: Navigate to Google Cloud Console.

<https://console.cloud.google.com/>



Step 2: For the first login, you have to click “I Understand”



Labs Access: Accessing Google Cloud Console

Step 3: You will be prompted to change password. Please enter a new password.



Step 4: Check the checkbox for “Terms of Service” and then proceed to click on “Agree and Continue”.

The screenshot shows the Google Cloud Console interface. At the top, there's a navigation bar with 'Cloud' selected, a search bar, and user information. Below the header, a banner offers '\$300 in free credits' for AI sandbox users. The main content area features a 'Welcome' message for 'AI Sandbox User 100' and several 'Popular getting started' sections with icons and descriptions. In the bottom right corner, a modal window is displayed, prompting the user to agree to the 'Google Cloud Platform Terms of Service' and 'Terms of Service' for specific services. The 'I agree to the Google Cloud Platform Terms of Service' checkbox is checked, and the 'AGREE AND CONTINUE' button is visible at the bottom of the modal.

Labs Access: Accessing Google Cloud Console

Proprietary + Confidential

Navigate to Vertex AI and then Workbench

Google Cloud

Ai-sandbox-Company88

Search (/) for resources, docs, products, and more

CREATE NEW REFRESH

LEARN

TOOLS

Vertex AI

Workbench

INSTANCES EXECUTIONS SCHEDULES

View: INSTANCES USER-MANAGED NOTEBOOKS MANAGED NOTEBOOKS

Workbench Instances have JupyterLab 3 pre-installed and are configured with GPU-enabled machine learning frameworks. [Learn more](#)

Instance name	Zone	Auto upgrade	Version	Machine Type	Gpus	Owner
ai-sb-user-523-gpu	asia-southeast1-a	-	M124	Graphics Optimized: 1 NVIDIA L4 GPU, 4 vCPUs, 16GB RAM	NVIDIA L4 x 1	391264778563-compute@developer.gserviceaccount.com
ai-sb-user-523-no-gpu	asia-southeast1-a	-	M124	Efficient Instance: 4 vCPUs, 16 GB RAM	None	391264778563-compute@developer.gserviceaccount.com
ai-sb-user-524-gpu	asia-southeast1-a	-	M124	Graphics Optimized: 1 NVIDIA L4 GPU, 4 vCPUs, 16GB RAM	NVIDIA L4 x 1	391264778563-compute@developer.gserviceaccount.com

COLAB Enterprise

Workbench

VERTEX AI STUDIO

BUILD WITH GEN AI

Extensions

DATA

Feature Store

Marketplace

https://336d31ca68426f-dot-asia-southeast1.notebooks.googleapisusercontent.com/?authuser=0&username=AI_Sandbox_Test_1

Labs Setup: Accessing Workbench Notebook

Step 4: Once at Workbench, ensure you are on the **INSTANCES** tab

Step 5: You should see that the notebooks created for your team.

Each user is assigned 2 notebooks. 1 with GPU and another without GPU.

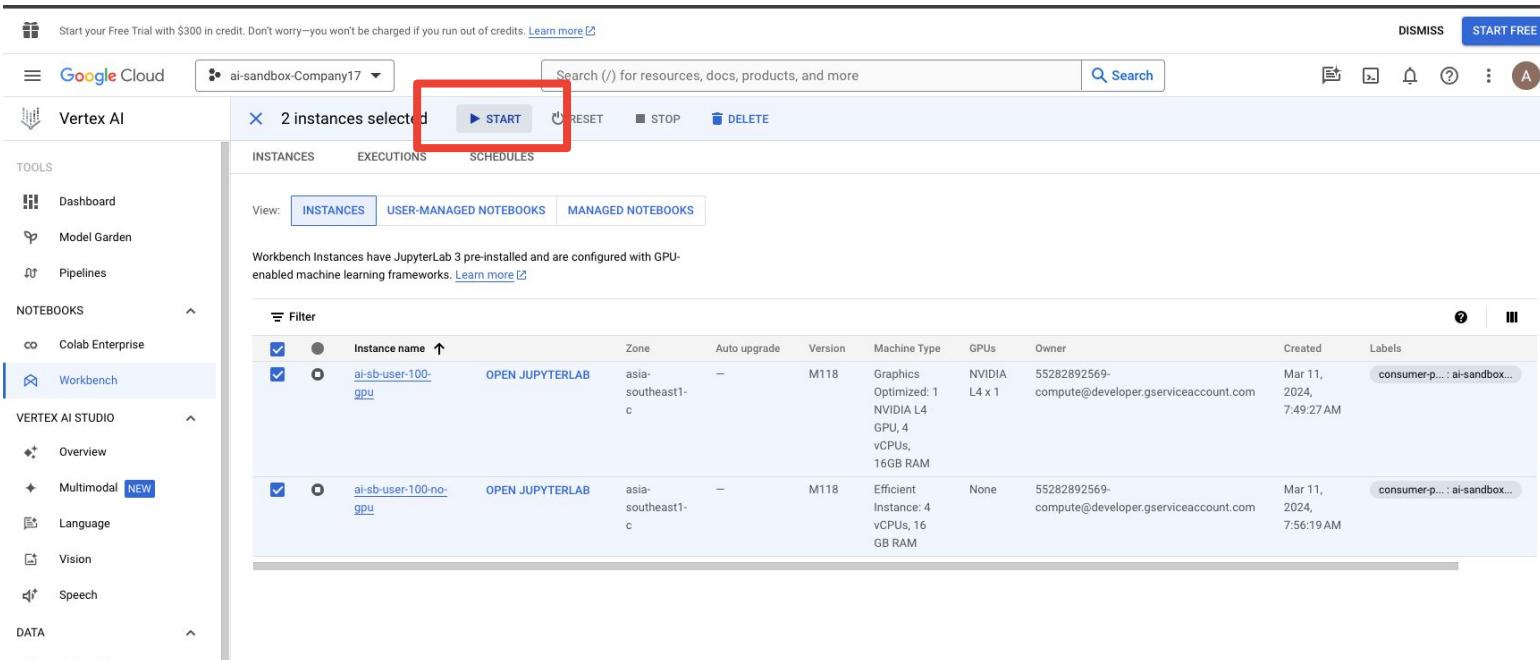
GPU instance: ai-gpu-100-XXX. Non-GPU instance: ai-no-gpu-100--XXX

The screenshot shows the Google Cloud Workbench Instances page. The left sidebar includes sections for Vertex AI, Tools (Dashboard, Model Garden, Pipelines), Notebooks (Colab Enterprise, Workbench), and Vertex AI Studio (Overview, Multimodal, Language, Vision, Speech). The main area has tabs for INSTANCES, EXECUTIONS, and SCHEDULES, with INSTANCES selected. A sub-header says "View: INSTANCES". Below this, a note states: "Workbench Instances have JupyterLab 3 pre-installed and are configured with GPU-enabled machine learning frameworks. [Learn more](#)". The main table lists two instances:

Instance name	Zone	Auto upgrade	Version	Machine Type	GPUs	Owner	Created	Labels
ai-gpu-100-3fb19a7	asia-southeast1-c	—	M118	Graphics Optimized: 1 NVIDIA L4 GPU, 4 vCPUs, 16GB RAM	L4 x 1	55282892569-compute@developer.gserviceaccount.com	Mar 11, 2024, 7:49:27 AM	consumer-p... : ai-sandbox...
ai-sb-user-100-no-gpu	asia-southeast1-c	—	M118	Efficient Instance: 4 vCPUs, 16 GB RAM	None	55282892569-compute@developer.gserviceaccount.com	Mar 11, 2024, 7:56:19 AM	consumer-p... : ai-sandbox...

Labs Setup: Accessing Workbench Notebook

Step 6: Select the 2 instances assigned and click on **START** in the top menu. If the instances are already started, skip to step 8.



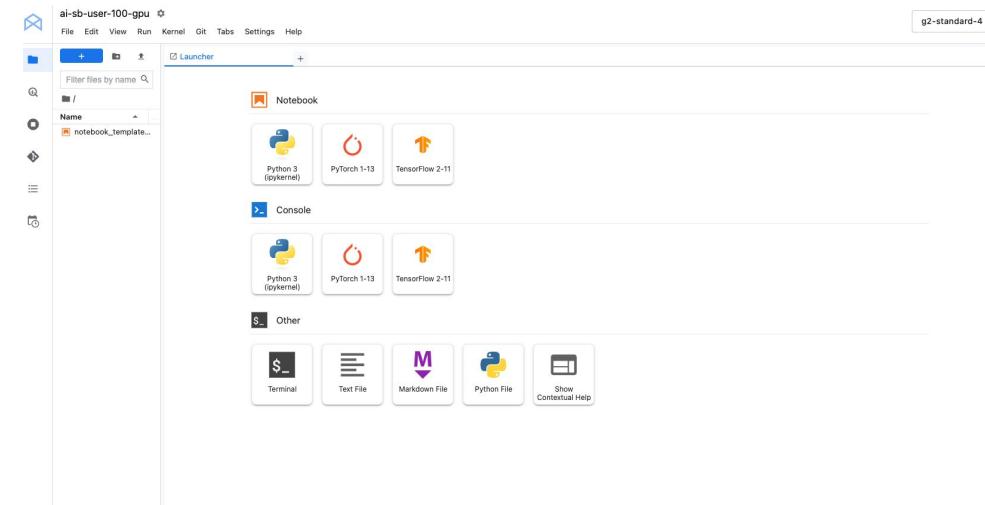
The screenshot shows the Google Cloud Vertex AI Workbench Instances page. A red box highlights the 'START' button in the top navigation bar, which is currently disabled (grayed out). The page displays two instances:

Instance name	Status	Zone	Auto upgrade	Version	Machine Type	GPUs	Owner	Created	Labels
ai-sb-user-100-gpu	OPEN JUPYTERLAB	asia-southeast1-c	—	M118	Graphics Optimized: 1	NVIDIA L4 GPU, 4 vCPUs, 16GB RAM	55282892569-compute@developer.gserviceaccount.com	Mar 11, 2024, 7:49:27 AM	consumer-p... : ai-sandbox...
ai-sb-user-100-no-gpu	OPEN JUPYTERLAB	asia-southeast1-c	—	M118	Efficient Instance: 4	vCPUs, 16 GB RAM	55282892569-compute@developer.gserviceaccount.com	Mar 11, 2024, 7:56:19 AM	consumer-p... : ai-sandbox...

Labs Setup: Accessing Workbench Notebook

Step 7: Wait 1-3 minutes for the instances to start. Verify that there is a green tick beside the instances.

Instance name ↑	Zone	Auto upgrade	Version	Machine Type	GPUs	Owner
ai-sb-user-100-gpu	asia-southeast1-c	-	M118	Graphics Optimized: 1 NVIDIA L4 x 1	NVIDIA GPU, 4 vCPUs, 16GB RAM	55282892569-compute@developer.gserviceaccount.com



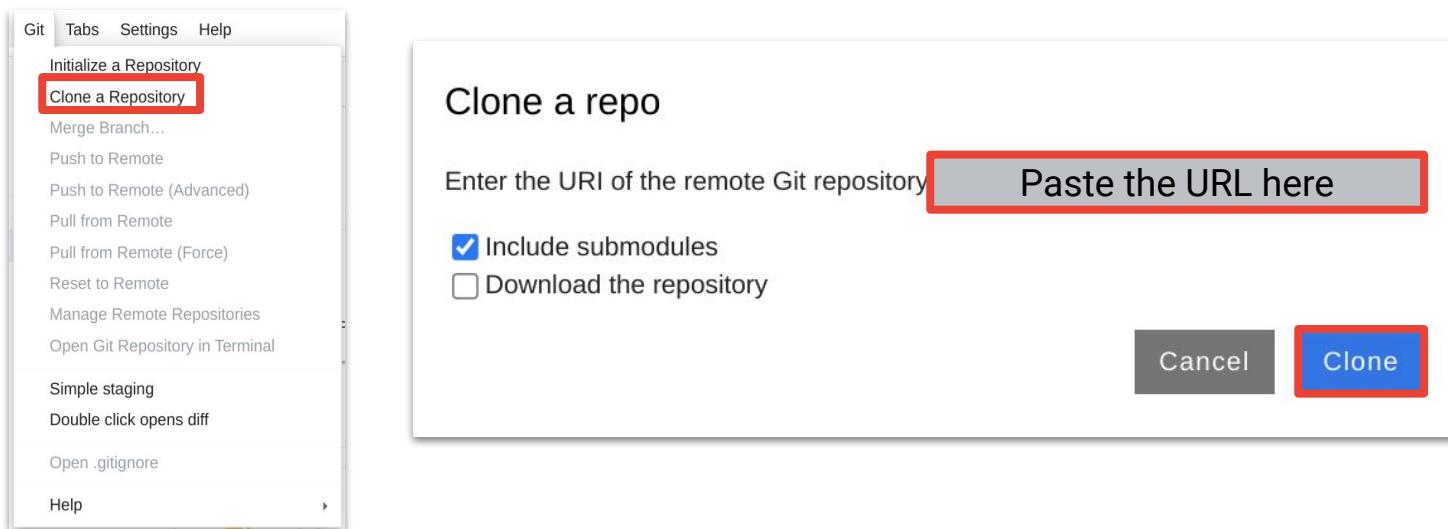
Step 8: Click on **OPEN JUPYTERLAB** for the **gpu instance** <ai-gpu-100-XXX>. A new tab will open with access to the Jupyter Notebook.

Labs Setup: Accessing Workbench Notebook

Step 9: Expand the **Git** menu and click on **Clone a Repository**

Step 10: Paste the URL provided for you into the text box and click **Clone**

URL = <https://github.com/analyticsrepo01/pru-ai-labs.git>



Labs Setup: Setting Up Notebooks

Step 12: Expand the **Run** menu and select **Run All Cells**

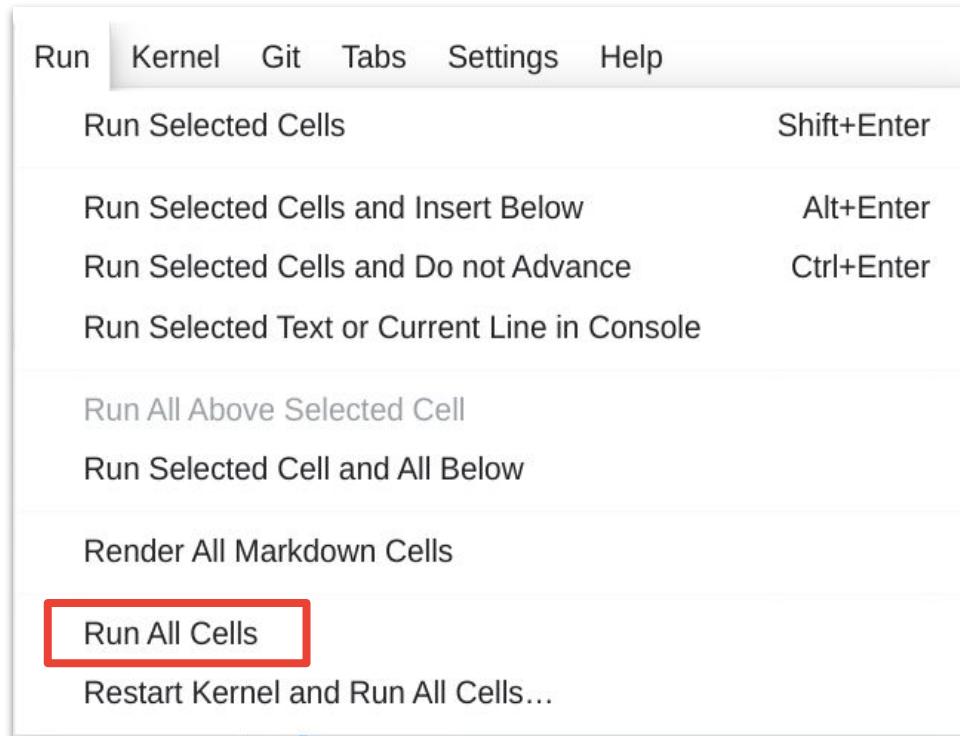


Image model with custom training

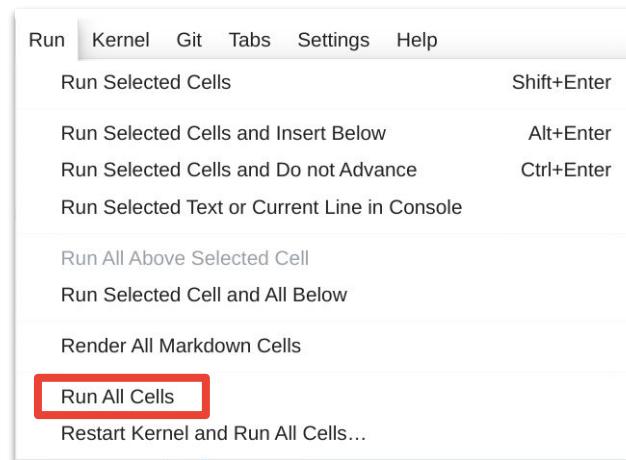
Labs Setup: Setting Up Notebooks

Step 11: Double click [pru_sd_xl_finetuning_dreambooth_lora.ipynb](#) to open the next notebook

- ❑ my_video.avi
- ❑ part_DIY_RAG_pdf.ipynb
- ❑ part_mixtral.ipynb
- ❑ [pru_sd_xl_finetuning_dreambooth_lora.ipynb](#)
- ❑ part5_RAG_VertexSearch.ipynb
- ❑ part6_Gemini_on_video.ipynb

Labs Setup: Setting Up Notebooks

Step 12: Expand the **Run** menu and select **Run All Cells**.



Open LLM with custom training

Labs Setup: Tuning a Open source LLM

Step 13: On Google Cloud Console, Navigate to Vertex AI → Model Garden

The screenshot shows the Google Cloud Vertex AI Model Garden interface. At the top, there's a navigation bar with 'Google Cloud' and a dropdown for 'Ai-sandbox-Company118'. A search bar contains the text 'Vertex AI', with a red box highlighting it. Below the search bar is a 'Dashboard' button. To the left is a sidebar titled 'TOOLS' containing 'Dashboard' (highlighted with a red box), 'Model Garden' (also highlighted with a red box), and 'Pipelines'. Under 'NOTEBOOKS', there are links for 'Colab Enterprise' and 'Workbench'. Under 'VERTEX AI STUDIO', there are links for 'Overview', 'Freeform', 'Chat', 'Vision', 'Translation', 'Speech', 'Prompt gallery', 'Prompt management', 'Training', and 'Marketplace'. The main content area features a 'Get started with Vertex AI' section with a 'ENABLE ALL RECOMMENDED APIs' button. It also includes a 'Tutorials' section with a 'VIEW TUTORIALS' button and three cards for 'Vertex AI Studio', 'Model Garden', and 'Colab Enterprise', each with a 'Try now' or 'Go to' button.

Google Cloud

Ai-sandbox-Company118

Vertex AI

Search

Dashboard

Get started with Vertex AI

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more about Vertex AI](#)

ENABLE ALL RECOMMENDED APIs

SHOW API LIST

Tutorials

Learn how to use generative AI, Gemini, and more on Vertex AI.

VIEW TUTORIALS

Vertex AI Studio

Prompt, test and tune generative AI models like Gemini.

Try now

Model Garden

Browse, customize, and deploy machine learning models. Choose from Google or popular open-source models.

Try now

Colab Enterprise

A new notebook experience with enterprise-grade privacy and security. Start coding in a couple clicks.

Go to Colab Enterprise

Dashboard

Model Garden

Pipelines

Colab Enterprise

Workbench

VERTEX AI STUDIO

Overview

Freeform

Chat

Vision

Translation

Speech

Prompt gallery

Prompt management

Training

Marketplace

TRY NOW

TRY NOW

TRY NOW

GO TO COLAB ENTERPRISE

Google



Vertex AI



Model Garden

EXPLORE GENERATIVE AI

VIEW MY ENDPOINTS & MODELS

DEPLOY FROM HUGGING FACE

VIEW RELEASE NOTES

TOOLS

Dashboard

Model Garden

Pipelines

NOTEBOOKS

Colab Enterprise

Workbench

VERTEX AI STUDIO

Overview

Freeform

Chat

Vision

Translation

Speech

Prompt gallery

Prompt management

Tuning

BUILD WITH GEN AI

Extensions

Marketplace

Modalities

Language	67
Vision	88
Tabular	7
Document	8
Speech	2
Video	6
Multimodal	21
Audio	1

Tasks

Generation	74
Classification	66
Detection	44
Extraction	28
Recognition	26
Translation	24
Embedding	7
Segmentation	12
Retrieval	2
Open vocabulary detection	2
Open vocabulary segmentation	2

Search models

Llama 3.1

Browse, customize, and deploy machine learning models with Model Garden. Choose from models created by Google and other providers.



Sort by: Trending Newest Last Update

Foundation models

→ SHOW ALL (93)

Pre-trained multi-task models that can be further tuned or customized for specific tasks.



Gemini 1.5 Pro

Created from the ground up to be multimodal (text, images, videos) and to scale across a wide range of tasks



Gemini 1.5 Flash

The best performing Gemini model with features for a wide range of tasks



Gemini 1.0 Pro

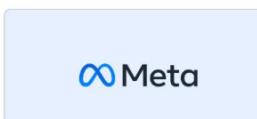
Designed to balance quality, performance, and cost for tasks such as content generation, editing, summarization, and classification



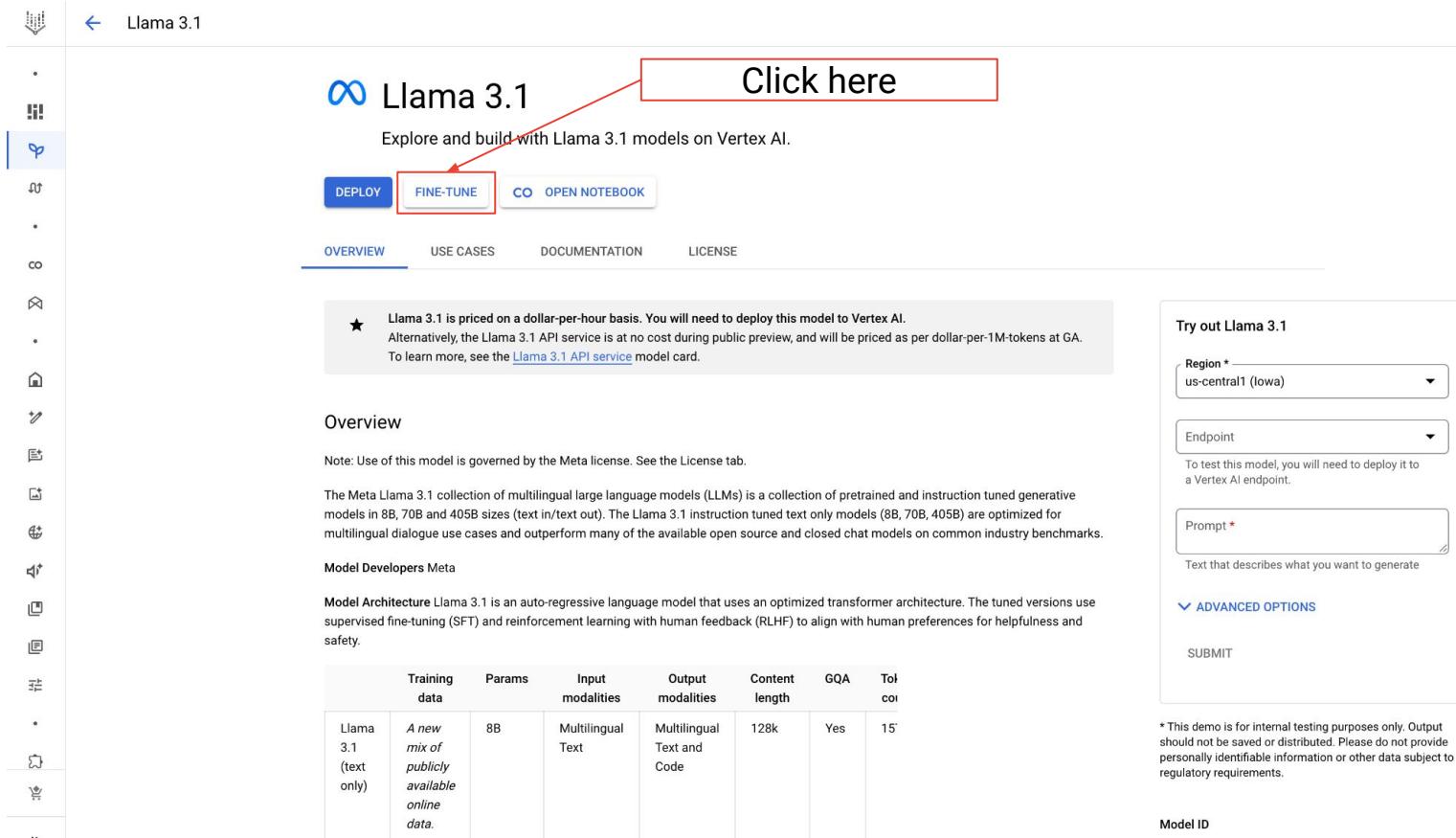
Gemini 1.0 Pro Vision

Created to be multimodal (text, images, code) and to scale across a wide range of tasks

Featured partners



Step 14: Choose Llama 3.1 model, Click on “FINE-TUNE”

A screenshot of the Llama 3.1 model page in Vertex AI. The page title is "Llama 3.1". Below it is a sub-header: "Explore and build with Llama 3.1 models on Vertex AI.". There are three main buttons at the top: "DEPLOY" (blue), "FINE-TUNE" (white with blue border), and "OPEN NOTEBOOK" (light blue). The "FINE-TUNE" button is highlighted with a red box and a red arrow pointing to it from the text "Click here" located above the button. Below these buttons are four tabs: "OVERVIEW" (underlined in blue), "USE CASES", "DOCUMENTATION", and "LICENSE". In the "OVERVIEW" section, there is a callout box containing the following text: ★ Llama 3.1 is priced on a dollar-per-hour basis. You will need to deploy this model to Vertex AI. Alternatively, the Llama 3.1 API service is at no cost during public preview, and will be priced as per dollar-per-1M-tokens at GA. To learn more, see the [Llama 3.1 API service](#) model card.

Click here

Llama 3.1

Explore and build with Llama 3.1 models on Vertex AI.

DEPLOY **FINE-TUNE** **OPEN NOTEBOOK**

OVERVIEW USE CASES DOCUMENTATION LICENSE

★ Llama 3.1 is priced on a dollar-per-hour basis. You will need to deploy this model to Vertex AI. Alternatively, the Llama 3.1 API service is at no cost during public preview, and will be priced as per dollar-per-1M-tokens at GA. To learn more, see the [Llama 3.1 API service](#) model card.

Overview

Note: Use of this model is governed by the Meta license. See the License tab.

The Meta Llama 3.1 collection of multilingual large language models (LLMs) is a collection of pretrained and instruction tuned generative models in 8B, 70B and 405B sizes (text in/text out). The Llama 3.1 instruction tuned text only models (8B, 70B, 405B) are optimized for multilingual dialogue use cases and outperform many of the available open source and closed chat models on common industry benchmarks.

Model Developers Meta

Model Architecture Llama 3.1 is an auto-regressive language model that uses an optimized transformer architecture. The tuned versions use supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align with human preferences for helpfulness and safety.

	Training data	Params	Input modalities	Output modalities	Content length	GQA	Tol coi
Llama 3.1 (text only)	A new mix of publicly available online data.	8B	Multilingual Text	Multilingual Text and Code	128k	Yes	15'

Try out Llama 3.1

Region * — us-central1 (Iowa)

Endpoint

To test this model, you will need to deploy it to a Vertex AI endpoint.

Prompt *

Text that describes what you want to generate

ADVANCED OPTIONS

SUBMIT

* This demo is for internal testing purposes only. Output should not be saved or distributed. Please do not provide personally identifiable information or other data subject to regulatory requirements.

Model ID

Create a fine-tuned model

Fine tune this model using supervised tuning.

Tuned model name *

Base model ?

Output directory * BROWSE ?
Input is required

Region * ?

The base model will be tuned with the following settings ([See pricing](#))

- Machine type: a2-ultragpu-8g
- Accelerator type: NVIDIA_A100_80GB
- Accelerator count: 8

Tuning parameters

Number of epochs * ?

Learning rate * ?

[VIEW TUNING CONFIG](#)

Dataset

Tuning dataset * BROWSE ?

Evaluation dataset * BROWSE ?

[VIEW EXAMPLE FORMAT](#)

START TUNING CANCEL

Create a bucket

Create or select the bucket

Get Started

Pick a globally unique, permanent name. [Naming guidelines](#)

Tip: Don't include any sensitive information

Optimize storage for data-intensive workloads ?

Labels (optional) ?

CONTINUE

- Choose where to store your data

Location: us (multiple regions in United States)
Location type: Multi-region
- Choose a storage class for your data

Default storage class: Standard
- Choose how to control access to objects

Public access prevention: On
Access control: Uniform
- Choose how to protect object data

Soft delete policy: Default
Object versioning: Disabled
Bucket retention policy: Disabled
Object retention: Disabled
Encryption type: Google-managed

CREATE CANCEL

Next start tining once all filled up

Vertex AI		Training		TRAIN NEW MODEL		IAM			
BUILD WITH GEN AI	Extensions	TRAINING PIPELINES		CUSTOM JOBS		IAM	IAM	IAM	IAM
		NAME	ID	STATUS	TYPE				
		llama3-1-lora-train-20241201-164595	233993907094945792	✓ Finished	Training pipeline	Custom	sec	11:34:27 AM	10:41:43 AM
		llama31-lora-20241201-143742	3612889896273838080	✓ Finished	Training pipeline	Custom	sec	28 min 8 sec	Dec 1, 2024, 11:05:52 PM
		llama3-1-lora-train-20241201-141259	3170974182838108160	🔴 Failed	Training pipeline	Custom	sec	52 min 45 sec	Dec 1, 2024, 11:05:45 PM
		llama31-lora-20241201-143729-7733063850	2477982790176473088	✓ Finished	Training pipeline	Custom	sec	28 min 8 sec	Dec 1, 2024, 11:05:39 PM
		llama3-1-lora-train-20241201-144150	3286941873242898432	✓ Finished	Training pipeline	Custom	sec	23 min 7 sec	Dec 1, 2024, 11:04:59 PM
		llama3-1-lora-train-20241201-134300	3813018604715114496	🔴 Failed	Training pipeline	Custom	sec	52 min 14 sec	Dec 1, 2024, 10:35:16 PM
		llama31-lora-20241201-143018	478384555623972864	🔴 Failed	Training pipeline	Custom	sec	1 min 32 sec	Dec 1, 2024, 10:31:52 PM
		llama3-1-lora-train-20241201-141611	4672924658566168576	✓ Finished	Training pipeline	Custom	sec	15 min 34 sec	Dec 1, 2024, 10:31:47 PM
		llama3-1-lora-train-20241201-141539	5396878298665975808	✓ Finished	Training pipeline	Custom	sec	15 min 35 sec	Dec 1, 2024, 10:15:40 PM
		llama3-1-lora-train-20241201-141534	4221438795922276352	✓ Finished	Training pipeline	Custom	sec	15 min 35 sec	Dec 1, 2024, 10:15:35 PM
		llama3-1-lora-train-20241201-142214	1870559790434877440	🔴 Failed	Training pipeline	Custom	sec	5 min 2 sec	Dec 1, 2024, 10:22:16 PM
		llama3-1-lora-train-20241201-141756	9148376788265598976	🔴 Failed	Training pipeline	Custom	sec	5 min 33 sec	Dec 1, 2024, 10:23:32 PM
		llama3-1-lora-train-20241201-131514	6548955378342690816	🔴 Failed	Training pipeline	Custom	sec	53 min 14 sec	Dec 1, 2024, 10:08:31 PM
		llama3-1-lora-train-20241201-134445	6964412443967619072	✓ Finished	Training pipeline	Custom	sec	16 min 35 sec	Dec 1, 2024, 10:01:23 PM
		llama3-1-lora-train-20241201-134438	3384050740208074752	✓ Finished	Training pipeline	Custom	sec	16 min 5 sec	Dec 1, 2024, 9:44:40 PM
		llama3-1-lora-train-20241201-122545	26206292010906600400	✓ Finished	Training	Custom	sec	12 min 4 sec	Dec 1, 2024, 9:25:47 PM
									Dec 1, 2024, 9:20:50 PM

Go to training tab - from left menu & check training started?

Example on custom data: Llama3.1 on Insurance data

Open and run all the notebook [pru_llama3_1_finetuning.ipynb](#)

The screenshot shows a Jupyter Notebook interface within a Google Cloud Colab environment. The left sidebar displays a file tree for the 'GenAI8 / pru-ai-labs/' directory, listing various notebooks and files. The main area shows the 'pru_llama3_1_finetuning.ipynb' notebook with its first code cell visible:

```
+ [20]: 1 # Copyright 2024 Google LLC
2 #
3 #
```

Below the notebook, a summary page titled "Vertex AI Model Garden - Llama 3.1 Finetuning Insurance Data" is displayed. It includes sections for "Overview", "Objective", "Costs", and "Before you begin".

Overview
This notebook demonstrates finetuning and deploying Llama 3.1 models with Vertex AI. After finetuning, we can deploy models on Vertex with GPU.

Objective

- Finetune Llama 3.1 models with Vertex AI Custom Training Jobs.
- Deploy finetuned Llama 3.1 models on Vertex AI Prediction.
- Send prediction requests to your finetuned Llama 3.1 models.

Costs
This tutorial uses billable components of Google Cloud:

- Vertex AI
- Cloud Storage

Before you begin

Install dependencies

```
[21]: 1 print("Installing google-cloud-aiplatform")
```

Run ALL

n1-standard-4

Filter files by name	
<input type="text"/>	
/ GenAI8 / pru-ai-labs /	
Name	Last Modified
sd_models	4 days ago
web-app	5 days ago
video	5 days ago
testing	5 days ago
images	5 days ago
backup_folder	5 days ago
pru_llama3_1_finetuning.ipynb	seconds ago
pru_sd_ll_finetuning_dreambooth_lor...	39 minutes ago
Multi-agent_002.ipynb	an hour ago
story.pdf	an hour ago
mdpdf.log	an hour ago
story.md	an hour ago
part6_Gemini_on_video.ipynb	12 hours ago
pru_DIY_RAG_pdf.ipynb	12 hours ago
pru_stable_diffusion_2_1.ipynb	3 days ago
pru_RAG_VertexSearch.ipynb	4 days ago
part_mixtral.ipynb	4 days ago
requirements.txt	5 days ago
questions_test.json	5 days ago
questions.json	5 days ago
part_llama3.ipynb	5 days ago
part_Imagen.ipynb	5 days ago
intro_multimodal_use_cases_latest.ip...	5 days ago
pru_qa.csv	5 days ago
Imagen_on_questions.ipynb	5 days ago
Agent_Example.ipynb	5 days ago
\$BUCKET_URI	5 days ago
Agent_Email_conversation002.invnb	5 days ago

```
Multi-agent_002.ipynb X Story_Multi_agent_002.ipynb X RAG_Crew.ipynb X pru_llama3_1_finetuning.ipynb X Terminal 1 X Terminal 2 X PyTorch 1-13 (Local)         Filter files by name  +            [20]: 1 # Copyright 2024 Google LLC  
2 #  
3 #
```

Vertex AI Model Garden - Llama 3.1 Finetuning Insurance Data

[Run in Colab Enterprise](#) [View on GitHub](#)

Overview

This notebook demonstrates finetuning and deploying Llama 3.1 models with Vertex AI. After finetuning, we can deploy models on Vertex with GPU.

Objective

- Finetune Llama 3.1 models with Vertex AI Custom Training Jobs.
- Deploy finetuned Llama 3.1 models on Vertex AI Prediction.
- Send prediction requests to your finetuned Llama 3.1 models.

Costs

This tutorial uses billable components of Google Cloud:

- Vertex AI
- Cloud Storage

Before you begin

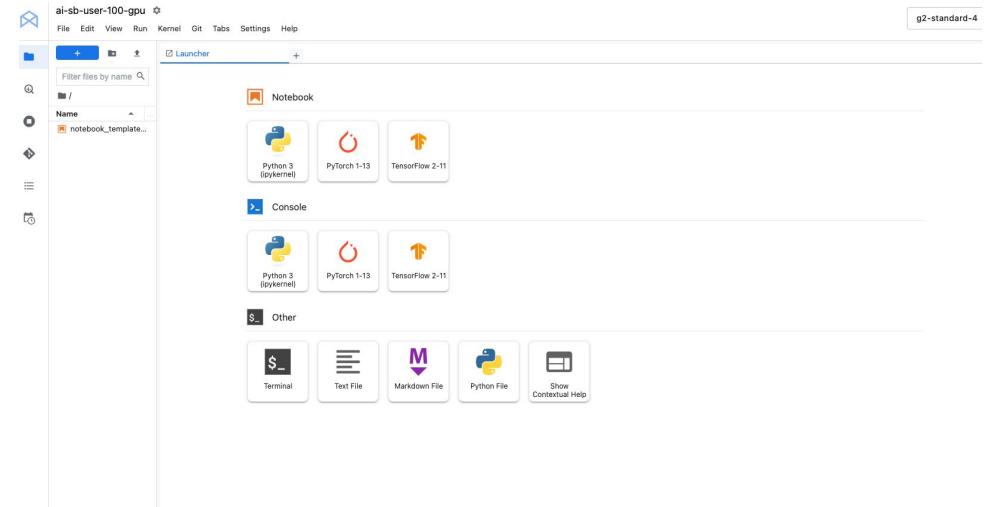
Install dependencies

```
[21]: 1 print("Installing google-cloud-aiplatform")
```

RAG for Prudential Use case

Labs Setup: Accessing Workbench Notebook

Step 15: Click on **OPEN JUPYTERLAB** for the **no-gpu** instance <ai-no-gpu-100-XXX>. A new tab will open with access to the Jupyter Notebook.

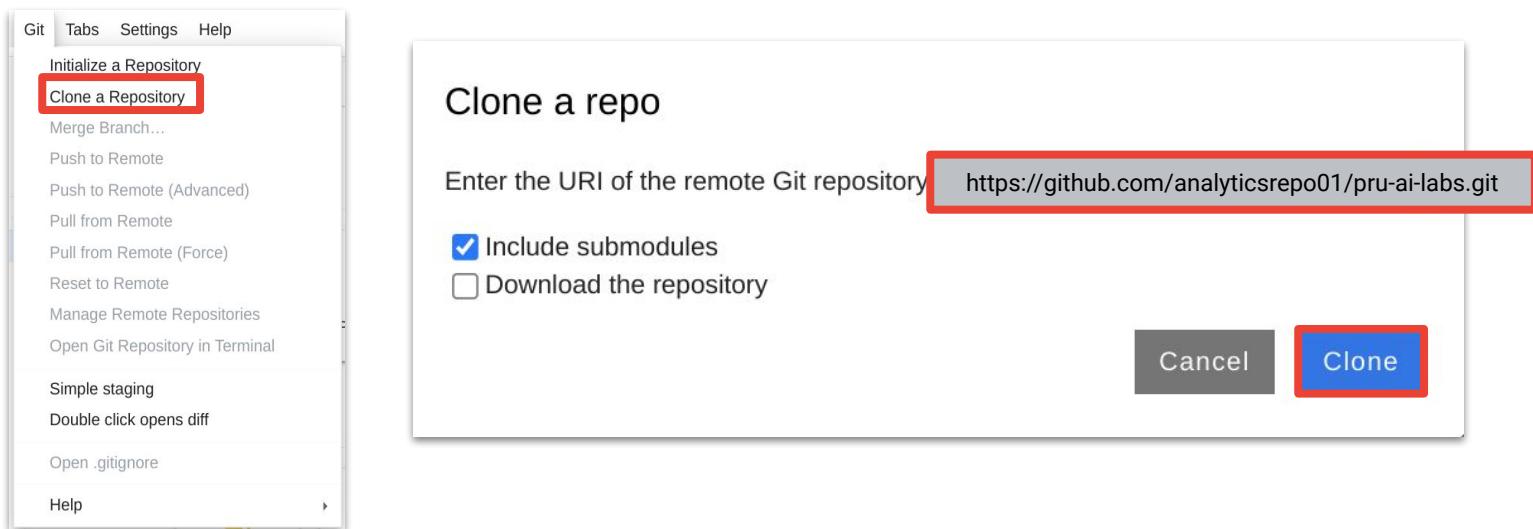


Labs Setup: Accessing Workbench Notebook

Step 16: Expand the **Git** menu and click on **Clone a Repository**

Step 17: Paste the URL provided for you into the text box and click **Clone**

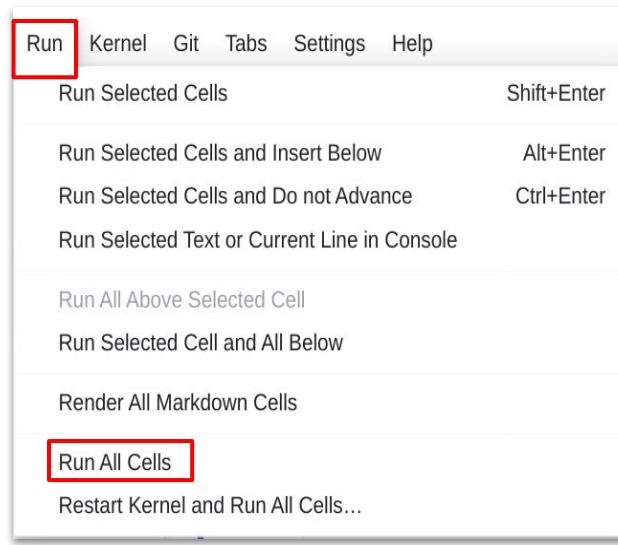
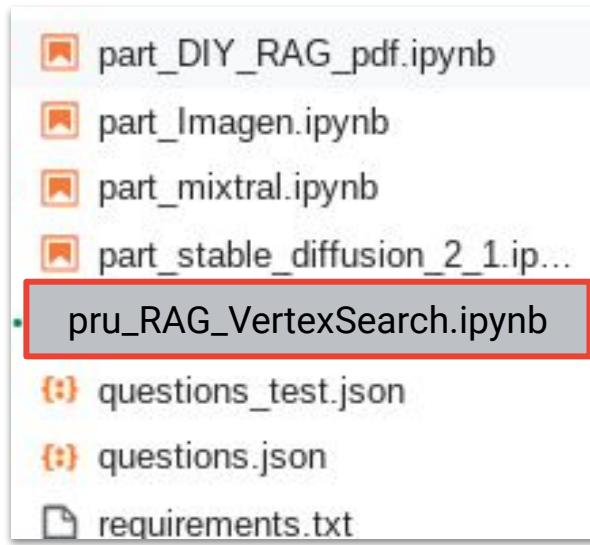
URL = <https://github.com/analyticsrepo01/pru-ai-labs.git>



Labs Setup: Vertex AI Search

Step 18: Double click **pru_RAG_VertexSearch.ipynb** to open the Jupyter Notebook

Step 19: In the top menu bar, expand the **Run** menu and select **Run All Cells**

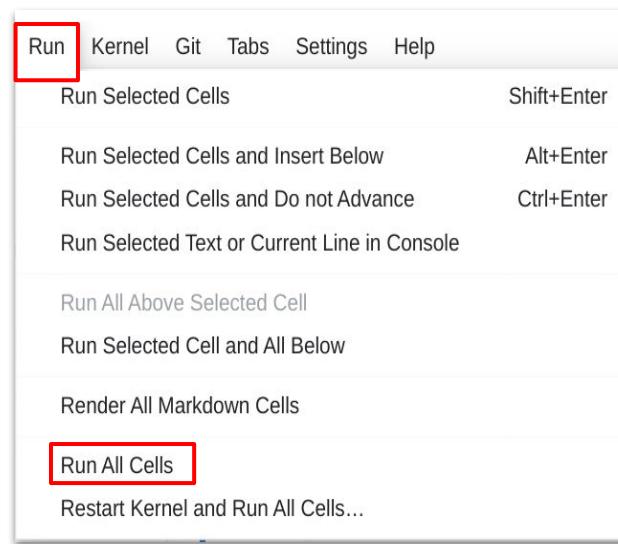
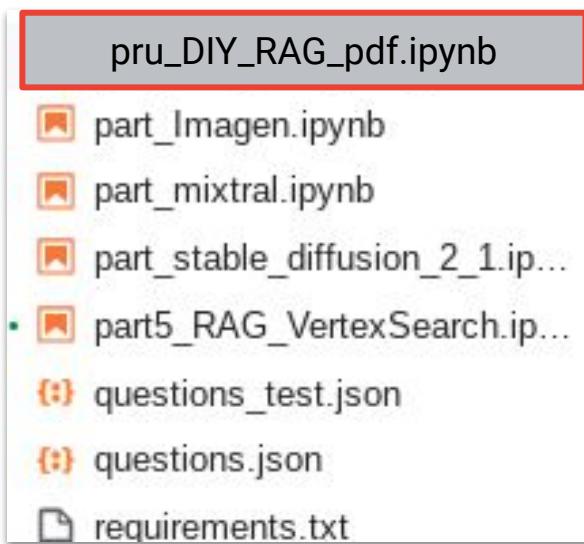


DIY RAG

Labs Setup: Vector AI Search

Step 20: Double click **pru_DIY_RAG_pdf.ipynb** to open the Jupyter Notebook

Step 21: In the top menu bar, expand the **Run** menu and select **Run All Cells**



Kernel Restarts after installation

Step1 : Click on this cell after restart

The screenshot shows a Jupyter Notebook interface running on a Google Colab instance. A red arrow points from the text "Step1 : Click on this cell after restart" to a specific cell in the notebook. Another red arrow points from the text "Step2 : Run Selected Cell and All Below" to the "Kernel" menu in the top navigation bar, which is open to show options like "Run All Above Selected Cell".

Step1 : Click on this cell after restart

Step2 : Run Selected Cell and All Below

Kernel

- Run All Above Selected Cell
- Run Selected Cell and All Below
- Render All Markdown Cells
- Run All Cells
- Restart Kernel and Run All Cells...

```

[1]: # Run this cell and all below

Declare Variables
[2]: #no spaces or special characters allowed), ensure that it is unique
import socket
import re

UNIQUE_PREFIX = socket.gethostname()

UNIQUE_PREFIX = re.sub('[^A-Za-z0-9]+', '', UNIQUE_PREFIX)

PREFIX_NUM_ONLY=int(str(re.search(r'\d+', UNIQUE_PREFIX).group()))

REGION_ALLOCATE=PREFIX_NUM_ONLY*3

[3]: PROJECT_ID = !(gcloud config get-value core/project)
PROJECT_ID = PROJECT_ID[0]

SVC_ACC = !(gcloud config get-value core/account)
SVC_ACC = SVC_ACC[0]

PROJECT_NUMBER=str(re.search(r'\d+', SVC_ACC).group())

LOCATION="asia-southeast1"

FOLDER_NAME="."

```

Multi Agents

ai-gpu-100-f20436e7

File Edit View Run Kernel Git Tabs Settings Help

g2-standard-4

File	
	+
	+
	+
	Filter files by name
/ pru-ai-labs /	
Name	Last Modified
■ images	a day ago
■ results	a day ago
■ testing	a day ago
■ vertex-ai-samples	a day ago
■ video	a day ago
■ web-app	a day ago
□ \$BUCKET_URI	a day ago
• Agent_Email_conversation002.ip... 2 minutes ago	
□ Agent_Example.ipynb	a day ago
□ Imagen_on_questions.ipynb	a day ago
□ intro_multimodal_use_cases_lat...	a day ago
□ mdpdf.log 6 minutes ago	
■ Multi-agent_002.ipynb seconds ago	
□ part_DIL_RAG_pdf.ipynb	a day ago
□ partImagen.ipynb	a day ago
□ part_llama3.ipynb	a day ago
□ part_mixtral.ipynb	a day ago
□ part6_Gemini_on_video.ipynb	a day ago
• pru_llama3_1_finetuning.ipynb 4 hours ago	
□ pru_qa.csv	a day ago
□ pru_RAG_VertexSearch.ipynb	a day ago
□ pru_sd_xl_finetuning_dreamboo...	a day ago
□ pru_stable_diffusion_2_1.ipynb	a day ago
□ questions_test.json	a day ago
□ questions.json	a day ago

Terminal 1 pru_llama3_1_finetuning.ipynb Agent_Email_conversation002.ipynb Multi-agent_002.ipynb

[1]: # !conda create -n crewai python=3.11
!conda activate crewai -
!pip install --upgrade google-cloud-aiplatform
!pip install -q -U 'crewai[tools]' mdpdf

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
gcsfs 2024.10.0 requires fsspec==2024.10.0, but you have fsspec 2024.9.0 which is incompatible.
kfp 2.5.0 requires kubernetes<27,>=8.0.0, but you have kubernetes 31.0.0 which is incompatible.
kfp 2.5.0 requires requests-toolbelt<1,>=0.8.0, but you have requests-toolbelt 1.0.0 which is incompatible.
kfp 2.5.0 requires urllib3<2.0.0, but you have urllib3 2.2.3 which is incompatible.

[2]: import re

PROJECT_ID = !(gcloud config get-value core/project)
PROJECT_ID = PROJECT_ID[0]

SVC_ACC = !(gcloud config get-value core/account)
SVC_ACC = SVC_ACC[0]

PROJECT_NUMBER=str(re.search(r'\d+', SVC_ACC).group())

LOCATION="asia-southeast1"

FOLDER_NAME=". "

[3]: from crewai import Agent, Task, Crew, Process
from crewai_tools import tool
from langchain_openai import ChatOpenAI
from crewai_tools.tools import FileReadTool
import os, requests, re, mdpdf, subprocess
from openai import OpenAI

[4]: !pip install --upgrade --quiet langchain-core langchain-google-vertexai
!pip install mdpdf

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.

Optional Lab :

Go to this github URL and clone

https://github.com/analyticsrepo01/ai_agents_v2

git clone <URL>



HANDS ON for setting up code assistant for
Notebooks

Go to this URL

<https://aider.chat/docs/install/install.html>

```
python -m pip install -U  
--upgrade-strategy only-if-needed  
aider-chat
```



Commands to use

```
git clone https://github.com/analyticsrepo01/gemini_openai_lib
```

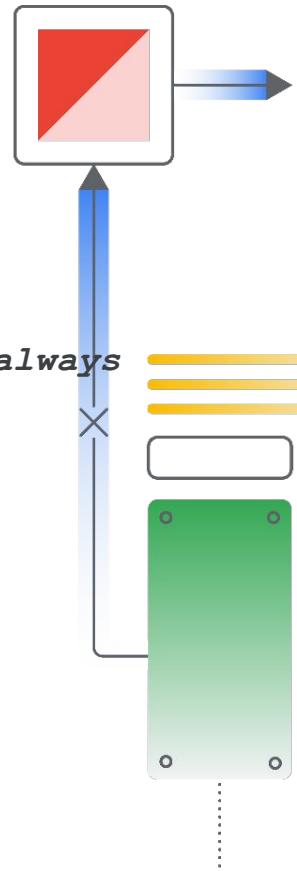
```
gcloud auth application-default login
```

USING GEMINI

```
aider --model vertex_ai/gemini-1.5-pro-002 -read Search.py --yes-always
```

USING Claude (optional)

```
aider --model vertex_ai/clause-3-5-sonnet-v2 --yes-always
```



Any Questions ? / Comments /
Thoughts ?

Labs Setup: Setting Up Notebooks

Note:

Avoid letting your laptops/computers enter sleep mode to prevent problems arising when running the notebooks

Thank you



Hands On Lab

11:30 - 12:15

- Lab: Multimodality with Gemini
- Lab: Imagen, Image Creation, Visual Captioning, Visual Question and Answer



Saurabh Mangal
AI Specialist CE

Google Partner + Google Engineers to support
the workshop

02

Gemini Labs

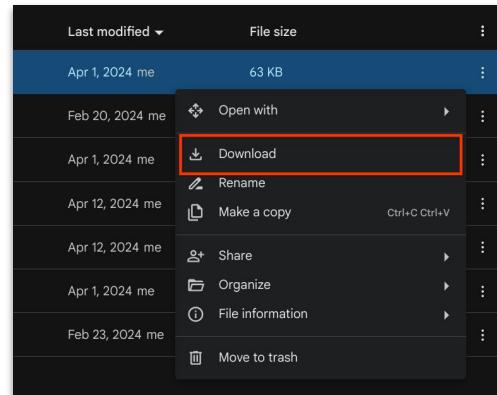
Gemini Lab: Saving Images/Videos

Step 1: Click on the link below to the Google Drive

<https://drive.google.com/drive/folders/1u78ByzEeqTTUGwdxBqBSCOfYnyrEETi0?usp=sharing>

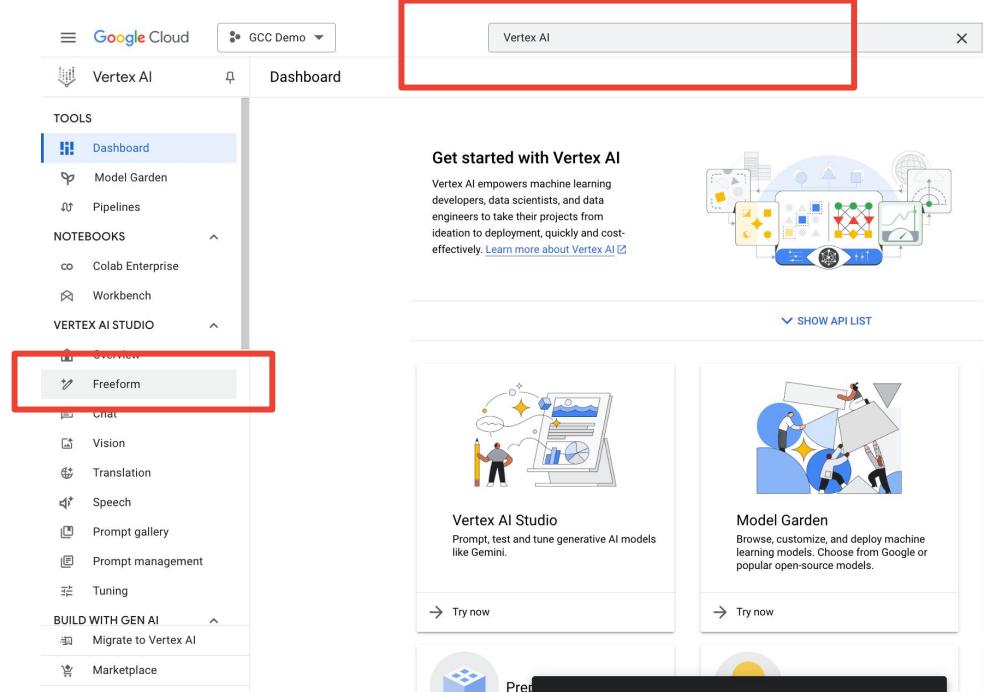
Step 2: Click on the three dots on the right side and select Download for each image/video

Type	People		Modified	⋮
Name	Owner	Last modified	File size	⋮
Gemini Lab 4-glasses2.png	me	Apr 1, 2024 me	63 KB	⋮
Gemini Lab1-fruit-basket.png	me	Feb 20, 2024 me	175 KB	⋮
Gemini Lab1-fruit-price.png	me	Apr 1, 2024 me	25 KB	⋮
Gemini Lab2-Oven.png	me	Apr 12, 2024 me	58 KB	⋮
Gemini Lab3-ER.png	me	Apr 12, 2024 me	30 KB	⋮
Gemini Lab4-glasses1.png	me	Apr 1, 2024 me	71 KB	⋮
mediterraneansea.mp4	me	Feb 23, 2024 me	6.4 MB	⋮



Gemini Pre-Lab: Navigating the Multimodal Console

Step 1: Navigate to **Vertex AI > Freeform** on the menu bar on the left hand side



Gemini Pre-Lab: Navigating the Multimodal Console

Step 2: Look for Prompt design (single turn) and click on “Open”

Multimodal

GET STARTED MY PROMPTS

Multimodal DOCUMENTATION API REFERENCE TUTORIALS

Prompt design (single turn)
Test the Gemini model using natural language, code, and an image.
Single-turn prompting has no conversation history.

Multi-turn prompt design (chat)
Test the model using natural language, code, and an image. Multi-turn prompting means the model tracks the conversation history and uses it as context for its responses.

OPEN

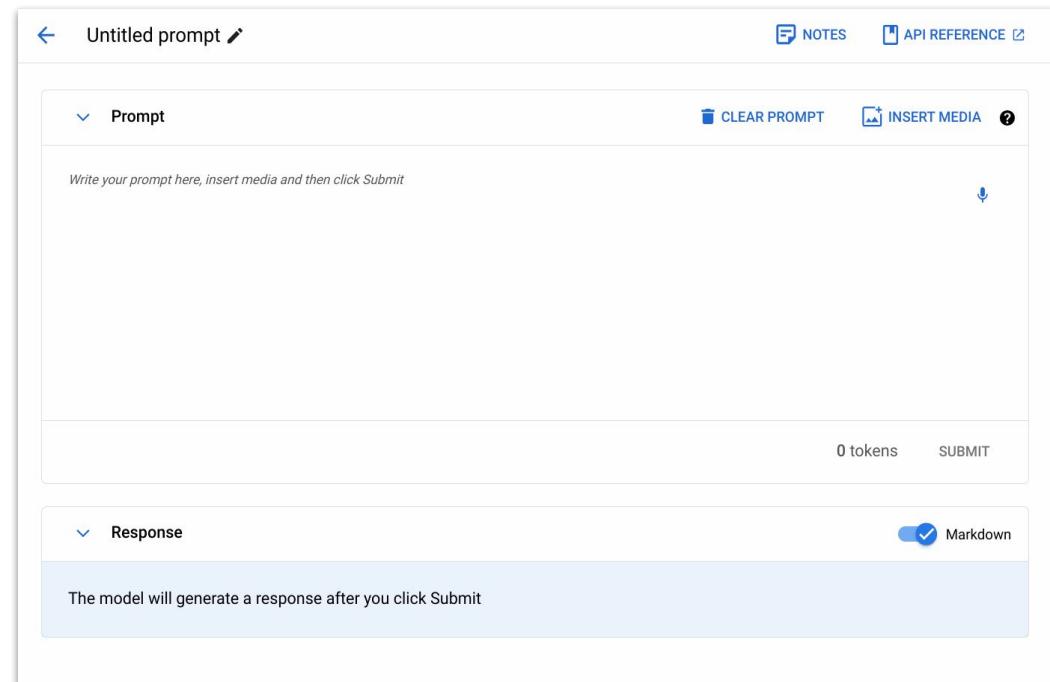
Gemini Pre-Lab: Navigating the Multimodal Console

Step 2: Click on “Insert Media”

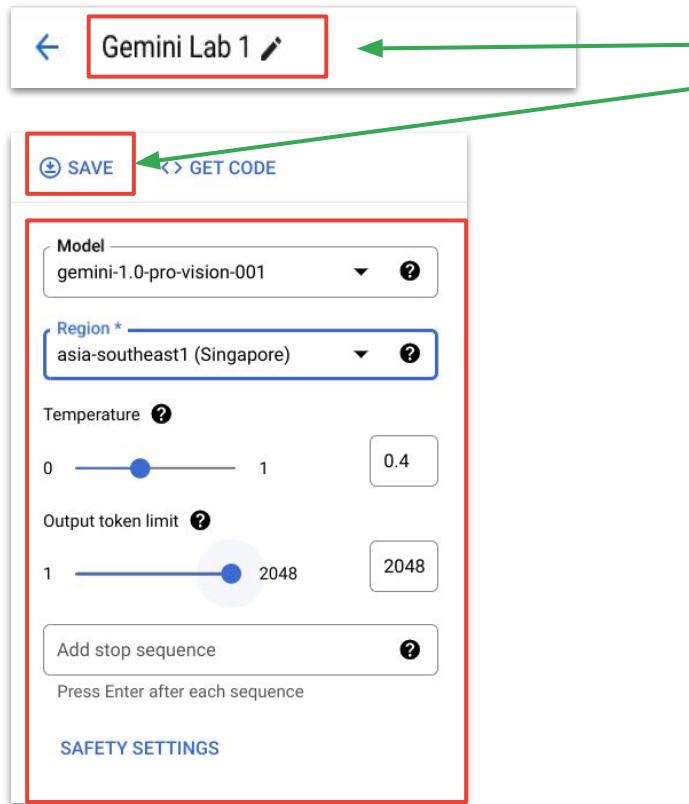
Step 3: Upload images that into the Prompt

Step 4: Include the prompt

Step 5: Click Submit



Gemini Pre-Lab: Gemini Parameters



Change the title and save the prompts for future reference

Feel free to adjust the Temperature and Output token limit to observe the different changes in output

- **Temperature:** Controls the degree of randomness in token selection. Lower temperatures results in a more accurate response, while higher temperatures can lead to more diverse or unexpected results.
- **Output token limit:** Restricts the length of the text an LLM can produce in the output response

Gemini Lab 1: Image understanding across multiple images

Prompt

How much should I pay for the fruits given the following price list?

Answer the question through these steps:

Step 1: Identify what kind of fruits there are in the first image.
Step 2: Count the quantity of each fruit.
Step 3: For each grocery in first image, check the price of the grocery in the price list.
Step 4: Calculate the subtotal price for each type of fruit.
Step 5: Calculate the total price of fruits using the subtotals.

Insert images:
Gemini Lab1-fruit-basket.png &
Gemini Lab1-fruit-price.png

Insert text prompts

1 tokens **SUBMIT**

Gemini Lab 1: Image understanding across multiple images

Response

Markdown

Step 1: The fruit basket contains apples and bananas.

Step 2: There are 3 apples and 2 bananas in the fruit basket.

Step 3: According to the price list: Apples cost \$1.50 each. Bananas cost \$0.80 each.

Step 4: The subtotal price for the apples is $3 \text{ apples} * \$1.50/\text{apple} = \4.50 . The subtotal price for the bananas is $2 \text{ bananas} * \$0.80/\text{banana} = \1.60 .

Step 5: The total price of the fruits is $\$4.50 + \$1.60 = \$6.10$.

Therefore, you should pay \$6.10 for the fruits in the basket.

Gemini Lab 2: Understanding Screens and Interfaces

Prompt

 CLEAR PROMPT

 INSERT MEDIA





Insert images:
Gemini Lab2-Oven.png

How can I reset the clock on this appliance?
Provide the instructions in English and then in French.

Insert text prompts

If instructions include buttons, also explain where those buttons are physically located

1 tokens

SUBMIT

Gemini Lab 2: Understanding Screens and Interfaces

Response



To reset the clock on this appliance:

Instructions in English

1. Locate the "Clock" button on the control panel. It is to the right of the digital display.
2. Press and hold the "Clock" button until the display flashes.
3. Use the number buttons to enter the correct time.
4. Press and hold the "Clock" button again to save the new time.

To reset the clock on this appliance:

Instructions in French

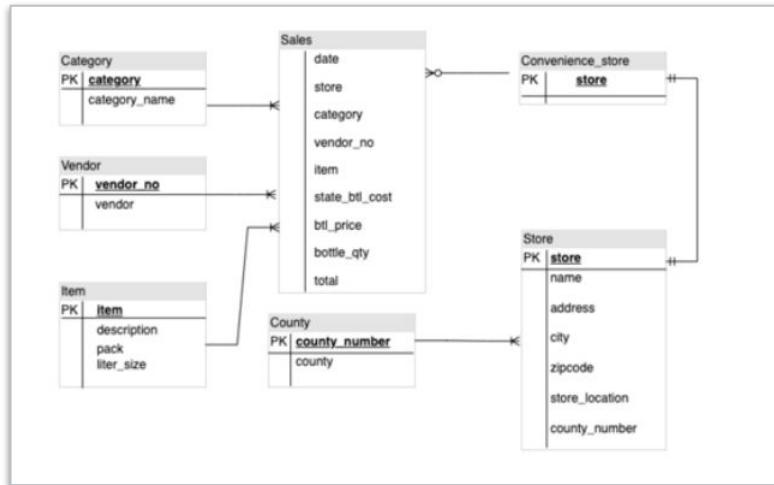
1. Locate the "Clock" button on the control panel. Il est à droite de l'affichage numérique.
2. Appuyez sur le bouton "Clock" et maintenez-le enfoncé jusqu'à ce que l'écran clignote.
3. Utilisez les boutons numériques pour entrer l'heure correcte.
4. Appuyez sur le bouton "Clock" et maintenez-le enfoncé à nouveau pour enregistrer la nouvelle heure.

Gemini Lab 3: Understanding ER in technical diagrams

Prompt

CLEAR PROMPT

INSERT MEDIA



Document the entities and relationships in this ER diagram



Insert text prompts

Insert images:
Gemini Lab3-ER.png

1 tokens

SUBMIT

Gemini Lab 3: Understanding Screens and Interfaces

Response



The entities in the ER diagram are:

- Category
- Vendor
- Sales
- Item
- County
- Convenience_store
- Store

The relationships in the ER diagram are:

- Category has many Sales
- Vendor has many Sales
- Sales has many Items
- Item belongs to one Vendor
- Item belongs to one Category
- Store has many Convenience_stores
- County has many Stores

Gemini Lab 4: Recommendations based on multiple images

Prompt CLEAR PROMPT INSERT MEDIA ?



Glasses 1 Glasses 2

Insert images:
Gemini Lab4-glasses1.png &
Gemini Lab4-glasses2.png

Which of these glasses you recommend
for me based on the shape of my face?
I have an oval shape face.

Explain how you reach out to this
decision.
Provide your recommendation based on
my face shape, and reasoning for each in
JSON format.

1 tokens SUBMIT

Gemini Lab 4: Recommendations based on multiple images

Response



```
{  
  "recommendation": "Glasses 1",  
  
  "reasoning": "Glasses 1 are a better fit for your oval face shape because they have a  
more angular frame that will help to balance out your features. Glasses 2 are a more  
rounded shape, which can be less flattering on oval faces."  
}
```

Gemini Lab 5: Generating a video description

▼ **Prompt**

 CLEAR PROMPT  INSERT MEDIA ?

What is shown in this video?
Where should I go to see it?
What are the top 5 places in the
world that look like this?

Insert video:
mediterraneansea.mp4

1 tokens **SUBMIT**

Insert text prompts

Gemini Lab 5: Generating a video description

Response



The video shows a beautiful coastline with a rocky cliff, a lighthouse, and a harbor. The water is crystal clear and there are some boats docked in the harbor. The sky is cloudy but still bright.

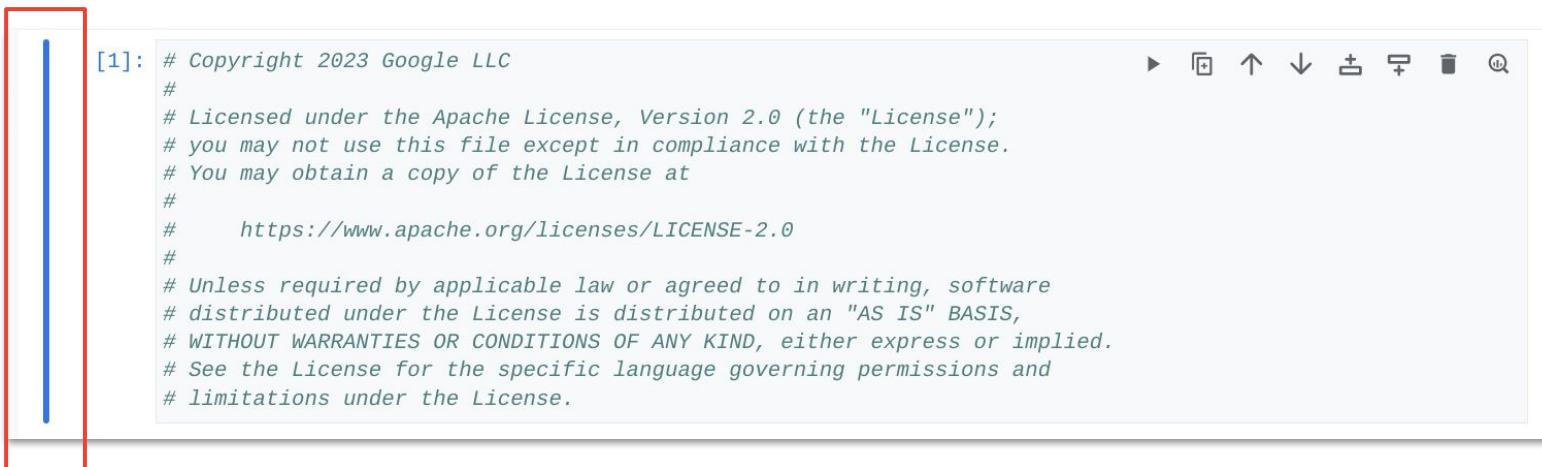
You can see this in **Antalya, Turkey**.

The top 5 places in the world that look like this are:

1. Amalfi Coast, Italy
2. Santorini, Greece
3. Cinque Terre, Italy
4. Positano, Italy
5. Dubrovnik, Croatia

Tutorial: Executing Cells in Jupyter Notebook

When a Jupyter Notebook is first opened, the 1st cell is selected by default. This is represented by a vertical blue bar on the left.



```
[1]: # Copyright 2023 Google LLC
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

Tutorial: Executing Cells in Jupyter Notebook

To execute a cell independently, you can either click the “play” button, or use the keyboard shortcut: **shift+enter**



```
[1]: # Copyright 2023 Google LLC
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

Tutorial: Executing Cells in Jupyter Notebook

Running a cell executes all the code within it and shows you the output (if any), and automatically selects the next cell. In this example, two lines are printed and the next cell is selected.

```
[2]: # GCP_PROJECT= PROJECT_ID=project_id= !(gcloud config get-value core/project)
PROJECT_ID= !(gcloud config get-value core/project)
PROJECT_ID = PROJECT_ID[0]
GCP_PROJECT = PROJECT_ID
LOCATION = REGION = 'asia-southeast1'

print(GCP_PROJECT)
print(PROJECT_ID)

ai-sandbox-company-18
ai-sandbox-company-18

[3]: import vertexai
from vertexai.generative_models import (
    GenerationConfig,
    GenerativeModel,
    Image,
    Part,
)

vertexai.init(project=PROJECT_ID, location=LOCATION)
```

Tutorial: Executing Cells in Jupyter Notebook

Occasionally, outputs may contain Warnings or Errors colour coded in red. These are generally harmless, but should your progress through the notebook be hindered, do check with any representative present.

```
from langchain_google_vertexai import VertexAIEMBEDDINGS  
  
embeddings = VertexAIEMBEDDINGS(model="textembedding-gecko@001")
```

WARNING:langchain_google_vertexai.embeddings:Model_name will become a required arg for VertexAIEMBEDDINGS starting from Feb-01-2024. Currently the default is set to textembedding-gecko@001

03

Imagen Labs (no-gpu instance)

Imagen Labs: Lab Setup

Step 1: Double click **part_Imagen.ipynb** to open the Jupyter Notebook

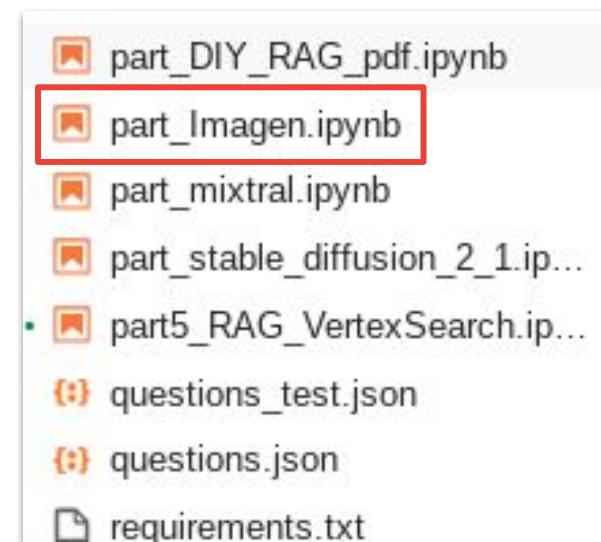
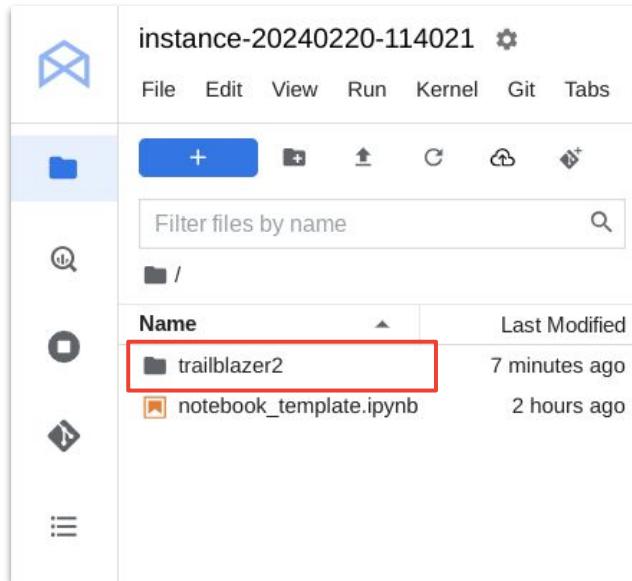


Imagen Labs: Lab Setup

Step 2: Read through the first cell to understand the objectives of this lab. Run through the cells one by one.

▶ 🔍 ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌃

Overview

Imagen on Vertex AI brings Google's state of the art generative AI capabilities to application developers. With Imagen on Vertex AI, application developers can build next-generation AI products that transform their user's imagination into high quality visual assets, in seconds.

With Imagen, you can do the following:

- Generate novel images using only a text prompt (text-to-image generation).
- Edit an entire uploaded or generated image with a text prompt.
- Edit only parts of an uploaded or generated image using a mask area you define.
- Upscale existing, generated, or edited images.
- Fine-tune a model with a specific subject (for example, a specific handbag or shoe) for image generation.
- Get text descriptions of images with visual captioning.
- Get answers to a question about an image with Visual Question Answering (VQA).

This notebook focuses on **image generation** only. You can read more about image generation feature from Imagen here.

Objectives

In this notebook, you will be exploring the image generation features of Imagen using the Vertex AI Python SDK. You will

- generate images using text prompts
- experiment with different parameters, such as:
 - increasing the number of images to be generated
 - fixing a seed number for reproducibility
 - influencing the output images using negative prompts

Imagen Labs: Text to Image

Step 3: Explore the different prompts and their outputs.

prompt = "A beautiful ice princess with silver hair, fantasy concept art style"



prompt = "A futuristic cityscape with towering skyscrapers and flying cars."



Google Cloud

Imagen Labs: Text to Image - Parameters

Step 4: Explore the parameter to customise the quantity of images generated.

```
prompt = "a delicious bowl of pho from Vietnam"

response = generation_model.generate_images(
    prompt=prompt,
    number_of_images=4,
)

display_images_in_grid(response.images)
```



number_of_images parameter can be set to generate up to 8 images

Imagen Labs: Text to Image - Parameters

Step 5: Explore the parameter to ensure the same images are generated each time

seed parameter can influence the model to create the same output from the same input every time (possibly in a different order)

```
response = generation_model.generate_images(  
    prompt=prompt,  
    number_of_images=4,  
    seed=42,  
)  
  
display_images_in_grid(response.images)
```

First Generation



```
response = generation_model.generate_images(  
    prompt=prompt,  
    number_of_images=4,  
    seed=42,  
)  
  
display_images_in_grid(response.images)
```

Second Generation



Imagen Labs: Text to Image - Parameters

Step 6: Explore the parameter to prevent unwanted objects from being generated

```
response = generation_model.generate_images(  
    prompt=prompt,  
    number_of_images=4,  
    seed=42,  
    negative_prompt="bean sprout",  
)  
  
display_images_in_grid(response.images)
```



negative_prompt parameter can exclude unwanted objects in the generated images

Imagen Labs: Introduction to Visual Captioning

Step 7: Ensure the image below is copied over from Github



- — — — —
- video
- web-app
- =1.29.0
- \$BUCKET_URI
- 360_F_284856166_H5I8...
- chkpt.mp3
- df_exploded_2.csv
- GettvImages-871168786
- google-cloud-next.jpeg**

Imagen Labs: Introduction to Visual Captioning

Step 8: Generate your first caption!

```
# Get a caption from the image
image_captioning_model.getCaptions(
    image=cloud_next_image,
)
```

```
['a group of people sitting in front of a google cloud next sign']
```

Imagen Labs: Visual Captioning - Parameters

Step 9: Explore the parameter to customise the quantity of captions generated

```
# Get 3 captions from the image
image_captioning_model.getCaptions(
    image=cloud_next_image,
    number_of_results=3,
)
['a group of people sitting in front of a google cloud next sign',
 'a group of people are sitting in front of a google cloud next sign',
 'a group of people sitting in front of a google cloud next stage']
```

number_of_results lets you generate up to
3 captions from a single image

Imagen Labs: Visual Captioning - Parameters

Step 10: Explore the parameter to customise the language of captions generated.

```
# Get 3 image captions in French
image_captioning_model.getCaptions(
    image=cloud_next_image,
    number_of_results=3,
    language="fr",
)
```

```
['un groupe de personnes assises devant un écran qui dit google cloud next',
 'un groupe de personnes assises devant un écran avec le logo google cloud next',
 'un groupe de personnes assises devant un écran de google cloud']
```

language lets you generate captions in a different language

Imagen Labs: Visual Captioning - Try It Yourself!

Step 11: Play around with the parameters and observe the different outputs



```
# Generate the visual captions for the image
image_captioning_model.getCaptions(
    image=user_image,
    number_of_results=3,
    language="en",
)
```

Imagen Labs: Visual Question & Answer

Step 12: Ask a question regarding the same image from the previous notebook



```
# Ask a question about the image
image_qna_model.ask_question(
    image=cloud_next_image, question="What is happening in this image?"
)
['google cloud next']
```

Imagen Labs: Visual Question & Answer

Step 13: Ask a different question and observe the output



```
# Ask a follow up question about the image
image_qna_model.ask_question(
    image=cloud_next_image, question="What are the people in the image doing?"
)
['sitting']
```

Imagen Labs: Visual Question & Answer - Parameters

Step 14: Explore the parameter to customise the quantity of answers generated

```
# Get 3 answers from the image
image_qna_model.ask_question(
    image=cloud_next_image,
    question="What are the people in the image doing?",
    number_of_results=3,
)
['sitting', 'watching', 'watching a presentation']
```

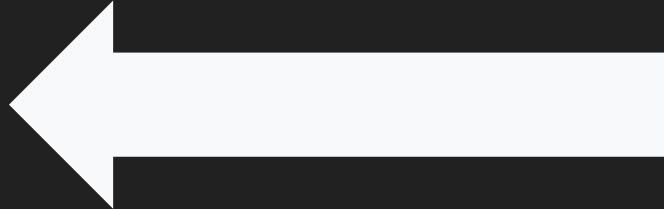
number_of_results lets you generate up to
3 answers from a single image/question

Imagen Labs: Visual Q & A - Try It Yourself!

Step 15: Try asking different questions and observe the outputs



Lunch time



**Resuming at 13:30 for hands-on
Continue the discussion during lunch**

Hands On Lab

13:00 - 14:30

- Kahoot!
- Lab: Vertex AI Search
- BigQuery ML



Saurabh Mangal
AI Specialist CE

Google Partners

Vertex AI Search (no-gpu instance)

Vertex AI Search: Overview of the steps

Step 1: Create a GCS bucket and upload the PDFs into the bucket

Step 2: Create an empty datastore

Step 3: Import documents from the GCS bucket into the datastore

Step 4: Create a Vertex Search engine and link it to the datastore

Step 5: Start querying your datastore through the search engine

Gemini powered search apps on Cloud Run

Here's what the output will look like:

The screenshot displays the Cymbal Search application interface. At the top, there is a navigation bar with links: SEARCH | DATASTORE | MULTI-TURN | CONTACT. To the right of the navigation bar is a user session identifier: searchtest_17210340. Below the navigation bar, the Cymbal Search logo is visible, featuring a stylized blue cloud icon and the text "CYMBAL SEARCH". On the left side of the interface, there are two dark-themed side panels. The top panel is titled "Filters" and contains a "Tenant" dropdown menu set to "All". The bottom panel is titled "Configurations" and contains a "# of Results per Page" dropdown menu set to "10". In the center of the interface is a light blue search bar with the placeholder text "Search Internal Documents" and a green search button containing a white magnifying glass icon.

Deployment of Models from Model Garden (gpu instance)

This is a review of the
[part_llama3_1_finetuning.ipynb](#) notebooks
that can be used from Model Garden

Llama 3.1: Prebuilt LLAMA 3.1 model with vLLM

Llama 3.1 is intended for commercial and research use in English. Instruction tuned models are intended for assistant-like chat, whereas pretrained models can be adapted for a variety of natural language generation tasks.

Step 1: Explore different prompts and outputs

prompt = "What is an apple?"

Prompt:

What is an apple?

Output:

It is a fruit. It is a fruit that grows on a tree. It is a fruit that grows on a tree that is called an apple tree. It is a fruit that grows on a tree that is called an apple tree that is grown in

Llama 3.1: Run inferences locally with prebuilt Llama 3.1 model - changing parameters

Step 2: Explore the parameter to customise the output.

Prompt: Text **prompt** that will be **generated**

Max tokens: **Word limit** of text generated

Temperature: **Randomness** of the text

Top P: **Probability** distribution

Top k: Restricts model to use the **top k words**

Raw Response: Whether to use a **raw response**

```
prompt = "what is an apple?" # @param {type: "string"}: Text prompt that Mistral will generate
max_tokens = 50 # @param {type:"integer"}: Word limit of text generated
temperature = 1.0 # @param {type:"number"}: Randomness of the text
top_p = 1.0 # @param {type:"number"}: probability distribution
top_k = 1 # @param {type:"integer"}: number of continuations to consider at each step when building the sequence
raw_response = False # @param {type:"boolean"}: whether to use raw response

# Overrides parameters for inferences.
# If you encounter the issue like `ServiceUnavailable: 503 Took too long to respond when processing`,
# you can reduce the maximum number of output tokens, such as set max_tokens as 20.
instances = [
    {
        "prompt": prompt,
        "max_tokens": max_tokens,
        "temperature": temperature,
        "top_p": top_p,
        "top_k": top_k,
        "raw_response": raw_response,
    },
]
```

Stable Diffusion: Introduction to Stable Diffusion V2.1

Step 1: Read through the first cell to understand the objectives of this lab.

▼ Overview

This notebook demonstrates running local inference for stabilityai/stable-diffusion-2-1 on either Colab or Vertex AI Workbench. This notebook also demonstrates finetuning stabilityai/stable-diffusion-2-1 with Dreambooth and deploying it on Vertex AI for online prediction.

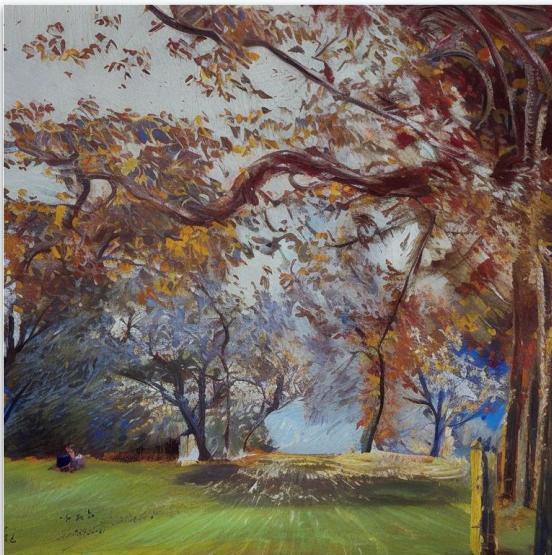
Objective

- Run local predictions for text-to-image and text-guided-image-to-image with serving dockers.
- Finetune the stabilityai/stable-diffusion-2-1 model with Dreambooth.
- Upload the model to Vertex AI Model Registry.
- Deploy the model to a Vertex AI Endpoint resource.
- Run online predictions for text-to-image and text-guided-image-to-image.

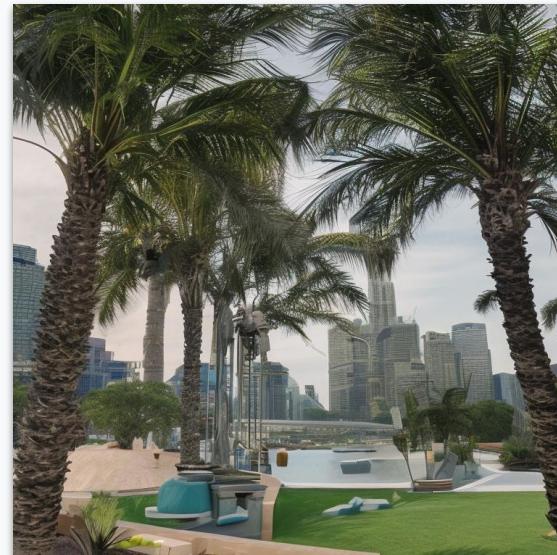
Stable Diffusion: Text to image

Text to image utilizes the stable diffusion model to generate an image from a prompt
Step 2: Explore the different prompts and their output

Prompt: Define **image to generate** from given input image



Prompt: “A peaceful park”



Prompt: “singapore city skyline”

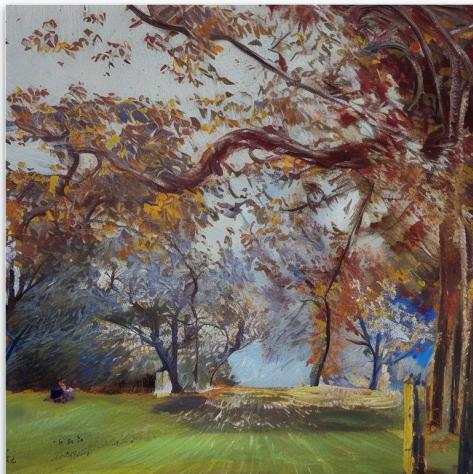
Stable Diffusion: Text-guided image to image

Img2Img takes a given image and edits the image according to the prompt

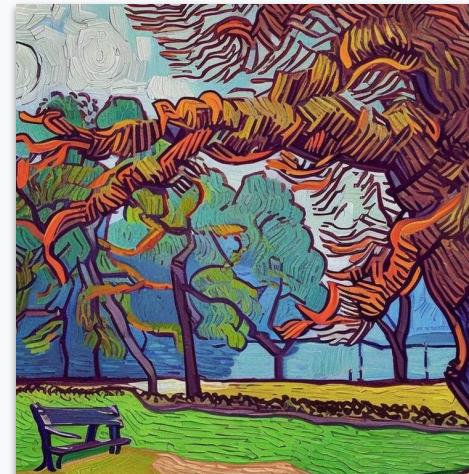
Step 3: Explore the different prompts and their output

Prompt: Define **image to generate** from given input image

Strength: **Variation** from input image from 0.0 to 1



Input image



Prompt: “A peaceful park in a van gogh style”
Strength: 0.4

Multi-Agent Lab

Introduction:

An agent is an autonomous unit programmed to:

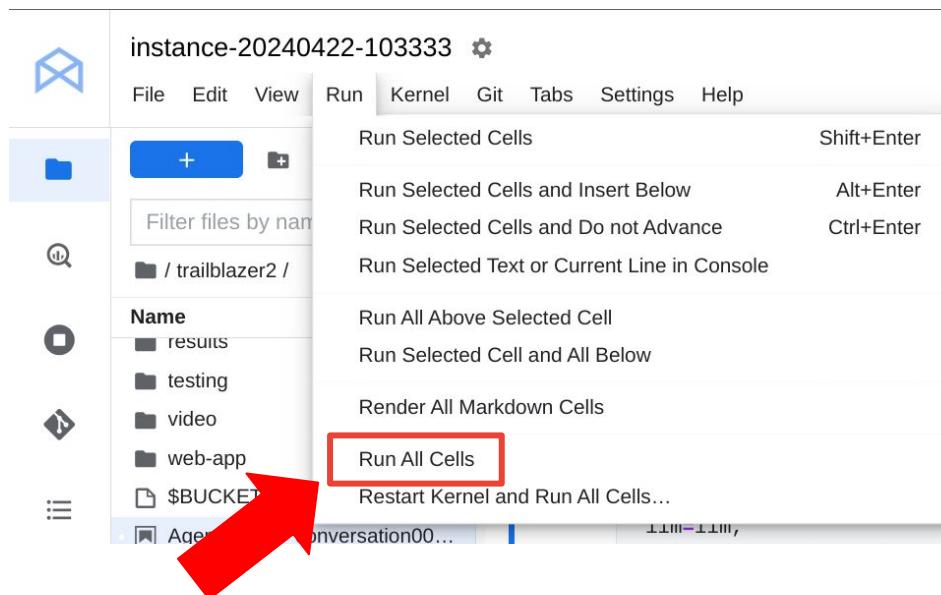
- Perform tasks
- Make decisions
- Communicate with other agents

```
aider --model vertex_ai/clause-3-5-sonnet-v2 --yes-always
```

Multi-Agent Lab: Crafting a Professional Email

Steps:

- Open [Agent_Email_conversation002.ipynb](#)
- Click Run > Run all cells



Multi-Agent Lab: Crafting a Professional Email

Cell output: Crafted email based on feedback

Hi [Name],

Are you ready to elevate your video editing skills and take them to the next level? Our comprehensive video editing solution is here to help you create stunning videos that will captivate your audience.

With our user-friendly interface and powerful features, you can:

- * **Effortlessly edit videos** with drag-and-drop functionality
- * **Add professional-looking effects** with just a few clicks
- * **Create engaging videos** that will leave a lasting impression

But don't just take our word for it. Here's what our satisfied customers have to say:

- * "This software is a game-changer. It's so easy to use, and I'm now able to create videos that look like they were made by a pro." - John Smith
- * "I highly recommend this editing solution. It's helped me save a lot of time and effort, and my videos now look amazing." - Jane Doe

Don't miss out on this opportunity to upgrade your video editing skills. Sign up for a free trial today and start creating videos that will wow your audience.

Sincerely,

[Your Name]

}Thought: I need to ask my coworker to generate two distinct variations of a cold email promoting a video editing solution.

Action: Delegate work to co-worker

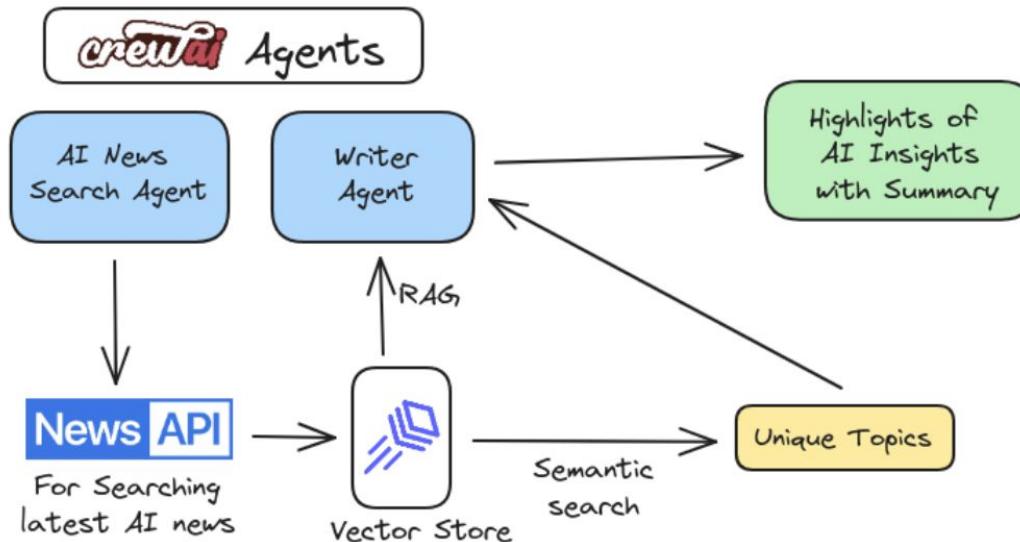
Action Input: {

"coworker": "Professional Email Author",
"task": "Generate two distinct variations of a cold email promoting a video editing solution.",
"context": "Please make sure that the emails are engaging and attention-grabbing. Use a strong subject line and a clear call to action. Also, ensure that the emails are grammatically correct and easy to read."
}

Multi-Agent Lab: Generating AI News Insights

Objectives:

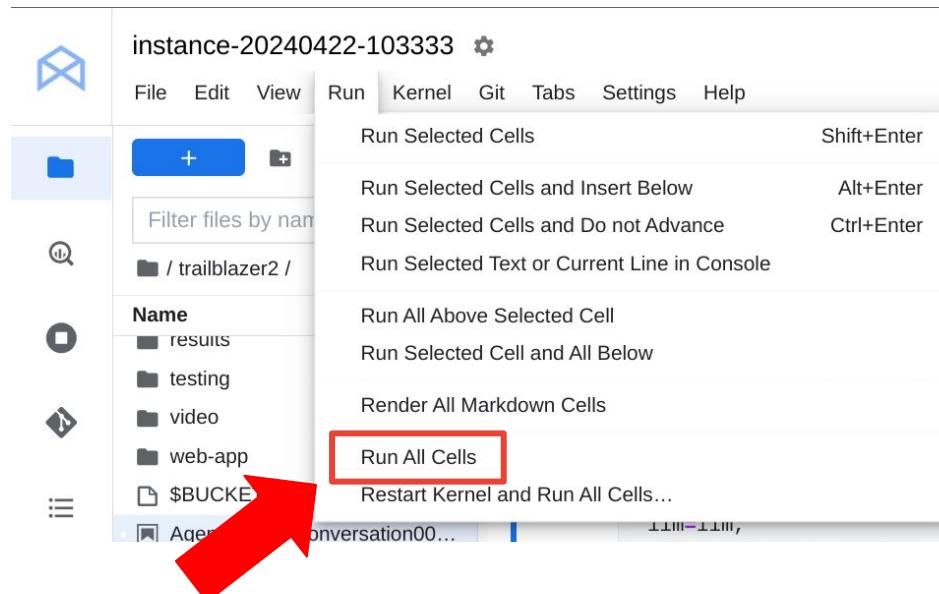
- Retrieve news about AI using RAG
- Generate summarized reports on AI trends



Multi-Agent Lab: Generating AI News Insights

Steps:

- Open Agent_Example.ipynb
- Click Run > Run all cells



Multi-Agent Lab: Generating AI News Insights

Cell output: Generated Summary Insights

```
print(result)
```

Key Trends:

- Generative AI models, such as ChatGPT, DALL-E 2, and Stable Diffusion, have continued to improve in quality and versatility, leading to their widespread adoption in various industries.

-Breakthrough Technologies:

- Quantum Computing, which has the potential to revolutionize various fields, including drug discovery, materials science, and financial modeling.

-Potential Industry Impacts:

- The integration of AI into the healthcare industry has the potential to improve patient outcomes and reduce healthcare costs.