UCLA Samueli School of Engineering

ECE C143A/C243A course project: Google cloud compute credits

Discussion - 6 (11/07/2025)

Shreeram Athreya

Discussion Goals

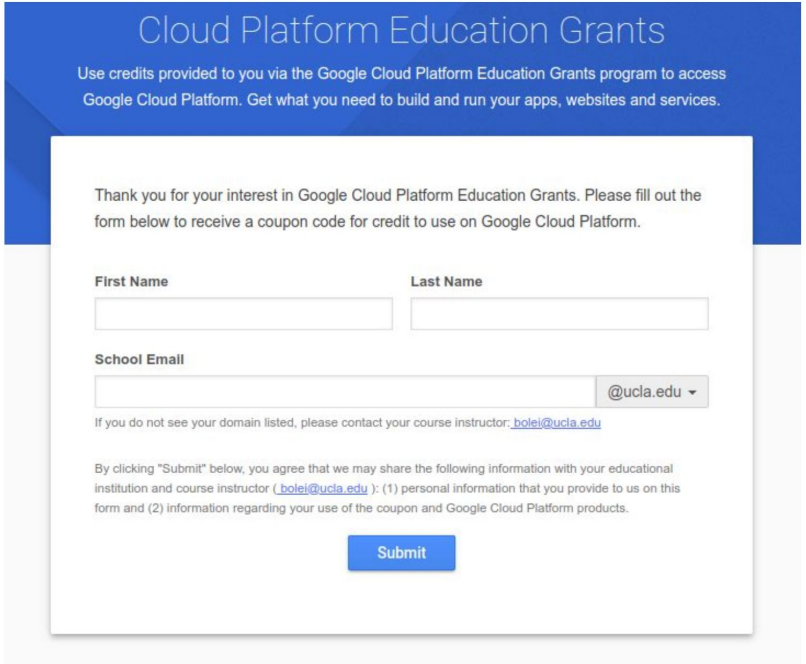
- Claim Google Cloud Credit
- Set up virtual machines (VMs) with GPUs using Google Cloud Credit
- Set up Google Colab to connect to VM as back-end

Claiming Google Cloud Credit

Google cloud credits: ECE C143A/C243A

- Google has provided us **Google Cloud Education Credits** worth $50 per coupon

- You can use Google Cloud VMs equipped with GPUs to run deep learning code, such as existing codebases available online, for your course project

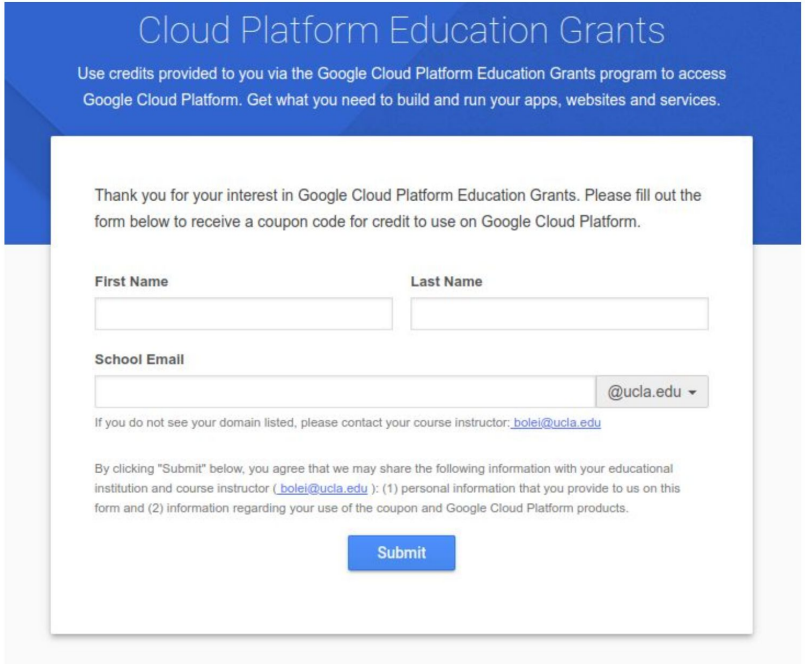


Retrieve your coupon using the link below

Here is the URL you will need to access in order to request a Google Cloud coupon. You will be asked to provide your school email address and name. An email will be sent to you to confirm these details before a coupon is sent to you.

Student Coupon Retrieval Link

- You will be asked for a name and email address, which needs to match your school domain (@ucla.edu / @g.ucla.edu). A confirmation email will be sent to you with a coupon code.
- You can request a coupon from the URL and redeem it until: **1/30/2026**
- Coupon valid through: **9/30/2026**
- You can only request ONE code per unique email address.



Important note on credit usage

- $50 seems like a lot of money, but in reality, this will probably last at most one week if you run your GPU VM continuously
- **Very important: Stop your VM when not in use to make your credits last longer!**

Important note on claiming coupon

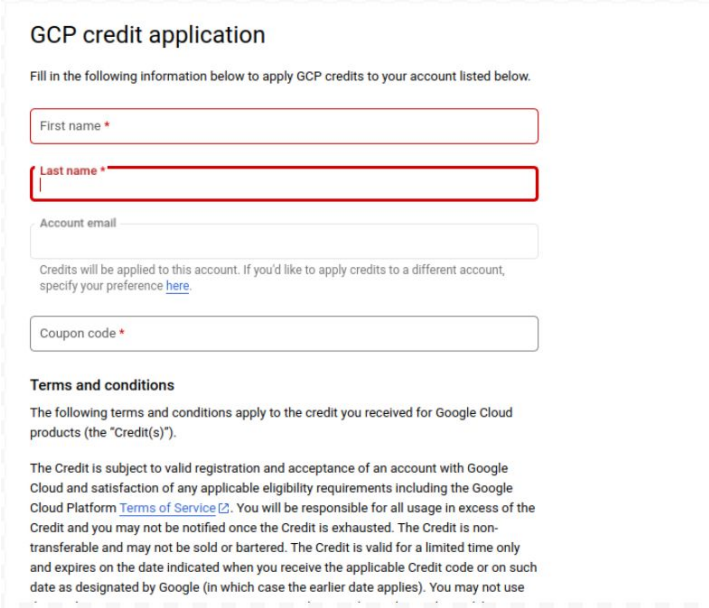
Please do not claim more than one credit coupon per person.

The number of coupons available is limited, only sufficient for 1 coupon per student. If you claim > 1 coupon, other students might not be able to claim any.

Redeeming Coupon

After submitting your coupon application, you will receive a link in your email with your unique coupon code.

Click on it and enter the necessary information

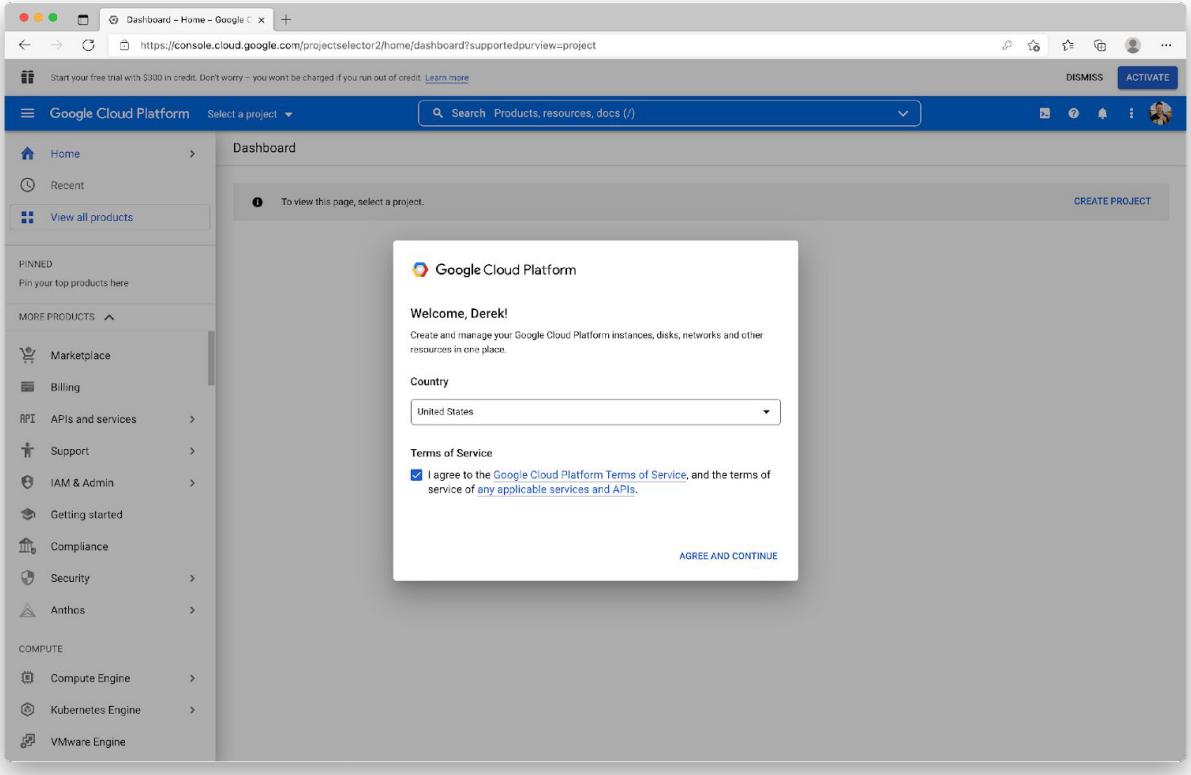


Create Instance with GPU on GCP

Slides partially taken from: Mingyu Derek Ma from CS188: Natural Language Processing

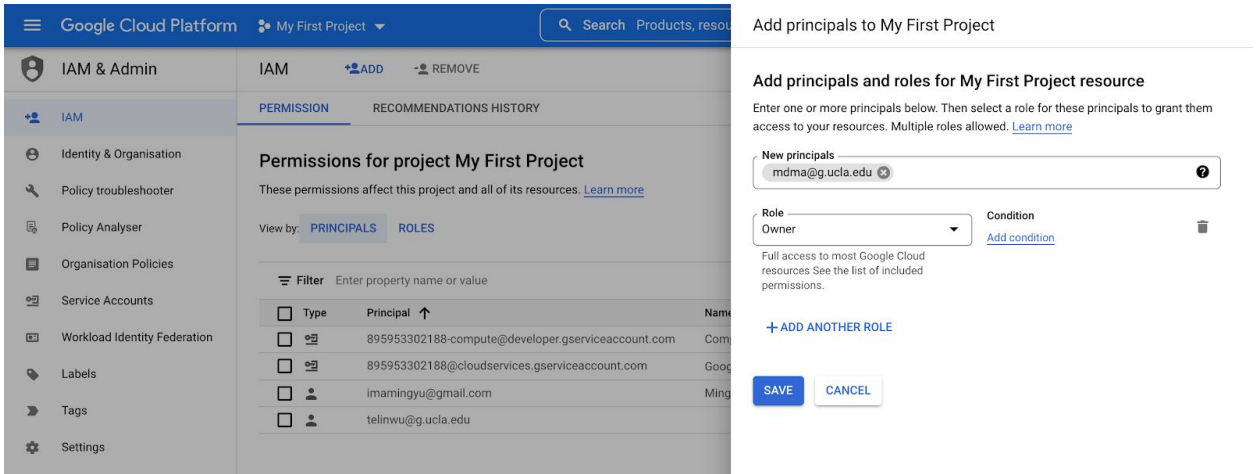
1. Create project

- Click ”Create Project”
- Or “New Project” after clicking the project name next to the “Google Cloud Platform” title
- New project may also be accessible after the organization (e.g. “g.ucla.edu”)



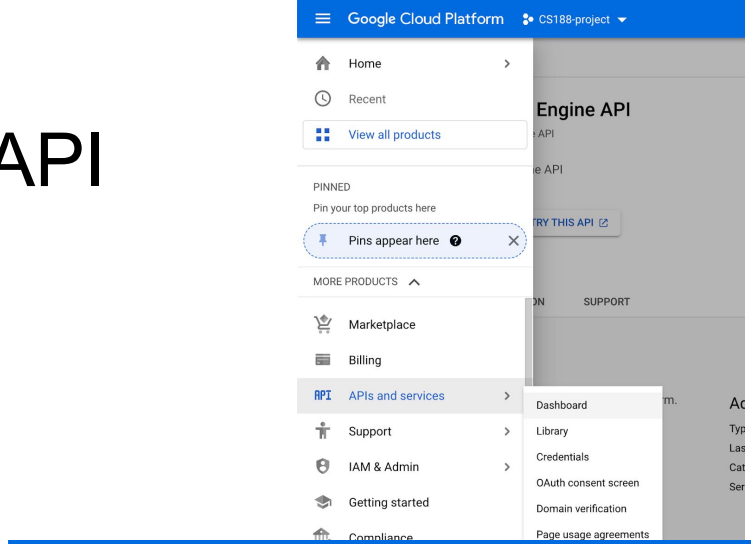
Share project with teammates

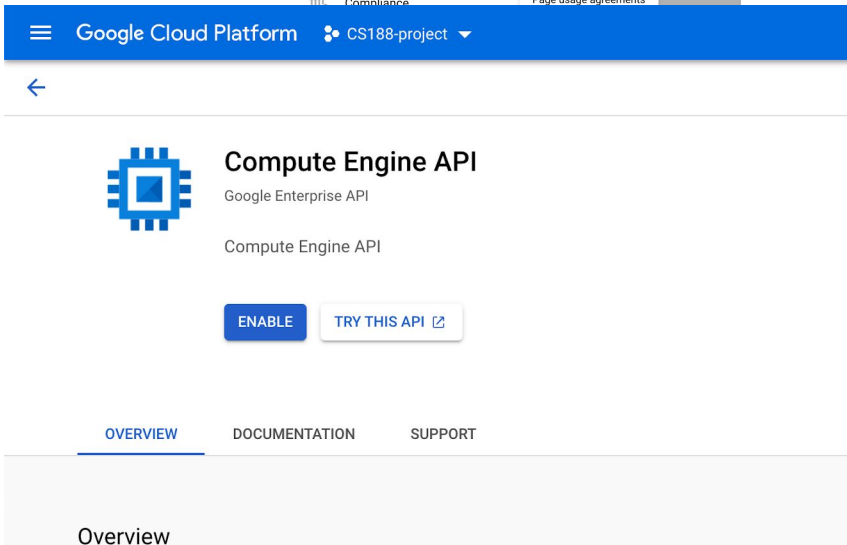
- *“IAM & Admin” > “IAM”*
- Add new user to the project, so other teammates can access the instances under this project



2. Enable Compute Engine API

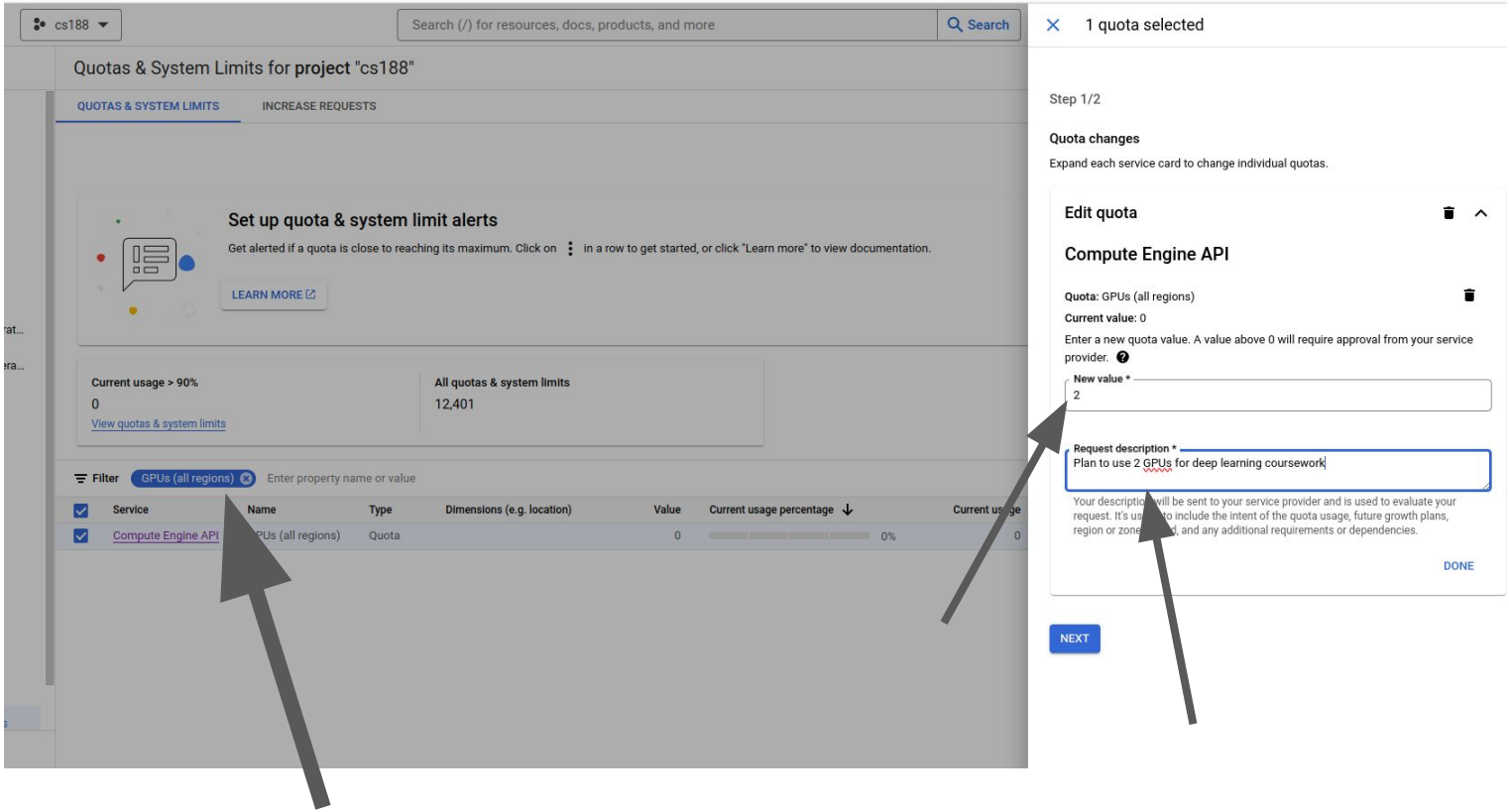
- It will prompt you to enable to the API when you first open the interfaces for Compute Engine
- Otherwise you can enable the API at *“API and services” > “Dashboard” > Search “Compute Engine API” > Enable*
  - This can take several minutes



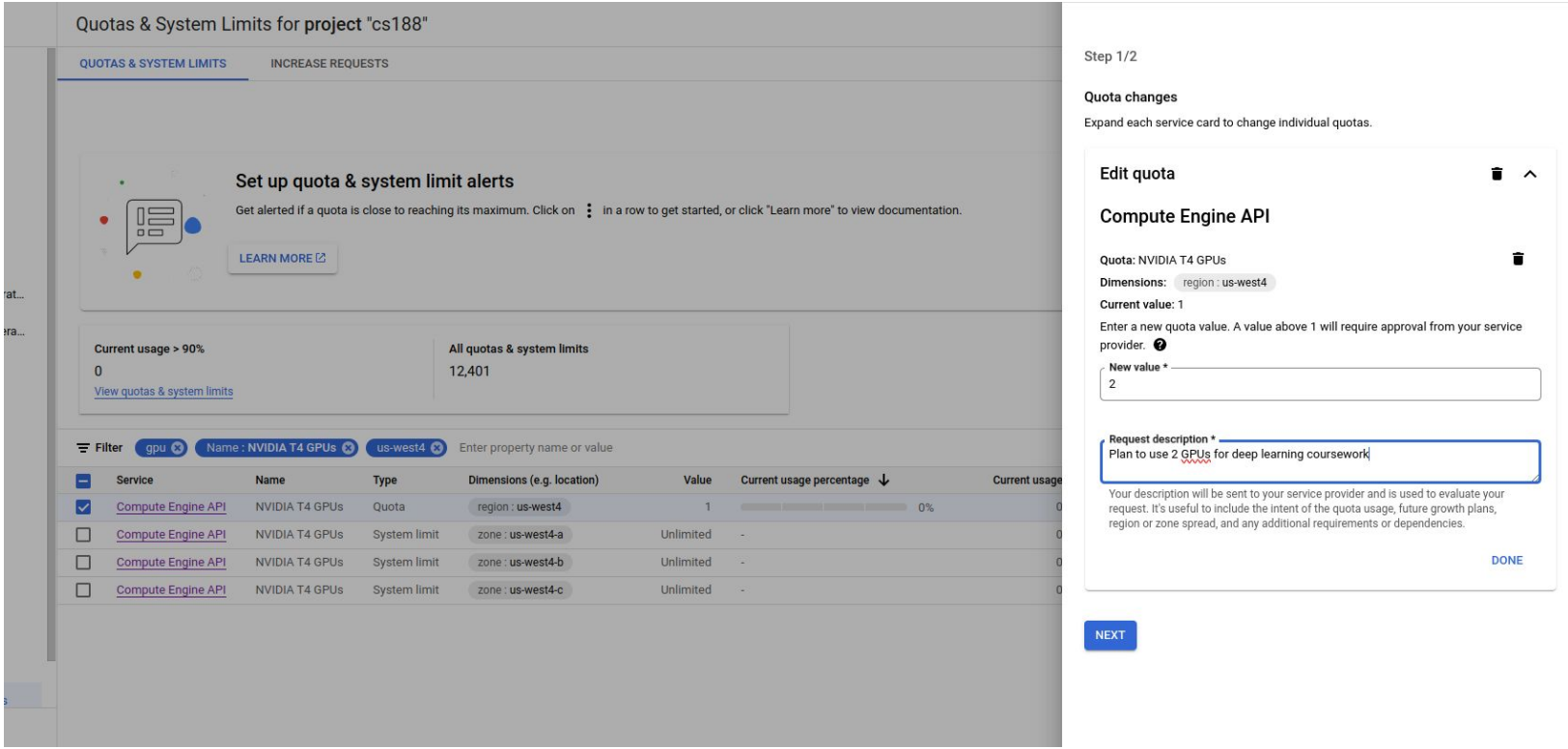


3. Check/change GPU quota after 48 hours

- By default, we can use 0 GPUs
- We need to request an increase in GPU quota
- Resource quotas | Compute Engine Documentation | Google Cloud
- Check quota at “IAM & Admin” > “Quotas”
- Add “GPUs (all regions)” in the filter
- Select quota item, click “Edit Quotas” (click the triple dots)
- Submit quota change request, need 24-48 hours to get response
    - Submit the quota increase request after 48 hours of creating your project, otherwise it will be declined
- us-west2 GPUs might already be available without making a quota request.



- Increase quota for *GPUs (all regions)*



- Increase quota for specific region and type of GPU you want to use (for example NVIDIA T4 GPUs at us-west4 is limited to 1 in the screenshot)

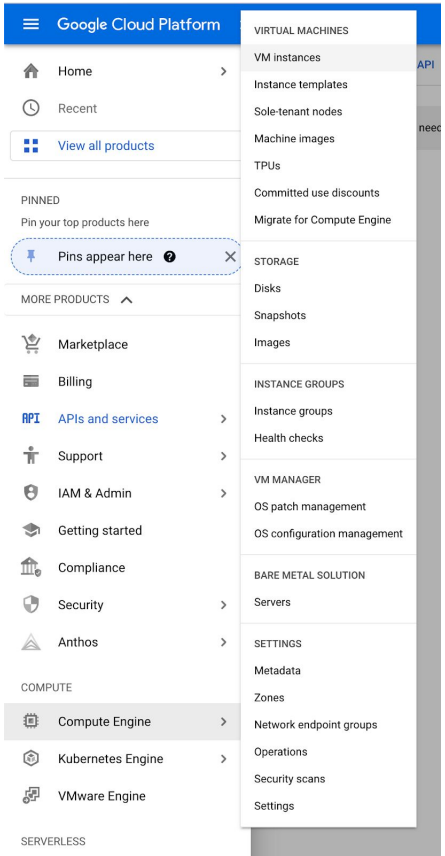
- [You can come back to this later after you have created your VM]

GPU Choices

- GPUs on Compute Engine | Compute Engine Documentation | Google Cloud
- GPUs pricing | Compute Engine: Virtual Machines (VMs) | Google Cloud
- GPU regions and zones availability | Compute Engine Documentation | Google Cloud

4. Create an instance with attached GPUs

- Enter “Compute Engine” > “VM Instances” > “Create Instance”



4. Create an instance with attached GPUs

- Create an instance
- Choose region and zone that has the GPU you requested
  - Check region supported GPU types in this link
  - For example, we choose “us-west4-b” to use NVIDIA T4 GPU
  - You must have an available quota in the region requested in order to create the instance
- Choose “GPU” under “Machine configuration”
- Select GPU type and number

Name *
instance-1

MANAGE TAGS AND LABELS

Region *
us-west4 (Las Vegas)
Region is permanent

Zone *
us-west4-b
Zone is permanent

Machine configuration

General purpose | Compute optimized | Memory optimized | ✓ GPUs

Graphics processing units (GPUs) accelerate specific workloads on your instances such as machine learning and data processing. Learn More

GPU type
NVIDIA T4

Number of GPUs
1

Enable Virtual Workstation (NVIDIA GRID)

	Series	Description	vCPUs	Memory	Platform
◉	N1	Balanced price & performance	1 - 96	1.8 - 624 GB	Intel Skylake

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a custom machine for your workload's particular needs. Learn more

PRESET | CUSTOM

n1-standard-1 (1 vCPU, 3.75 GB memory)

vCPU	Memory
1	3.75 GB

ADVANCED CONFIGURATIONS

Monthly estimate

$217.51

That's about $0.30 hourly

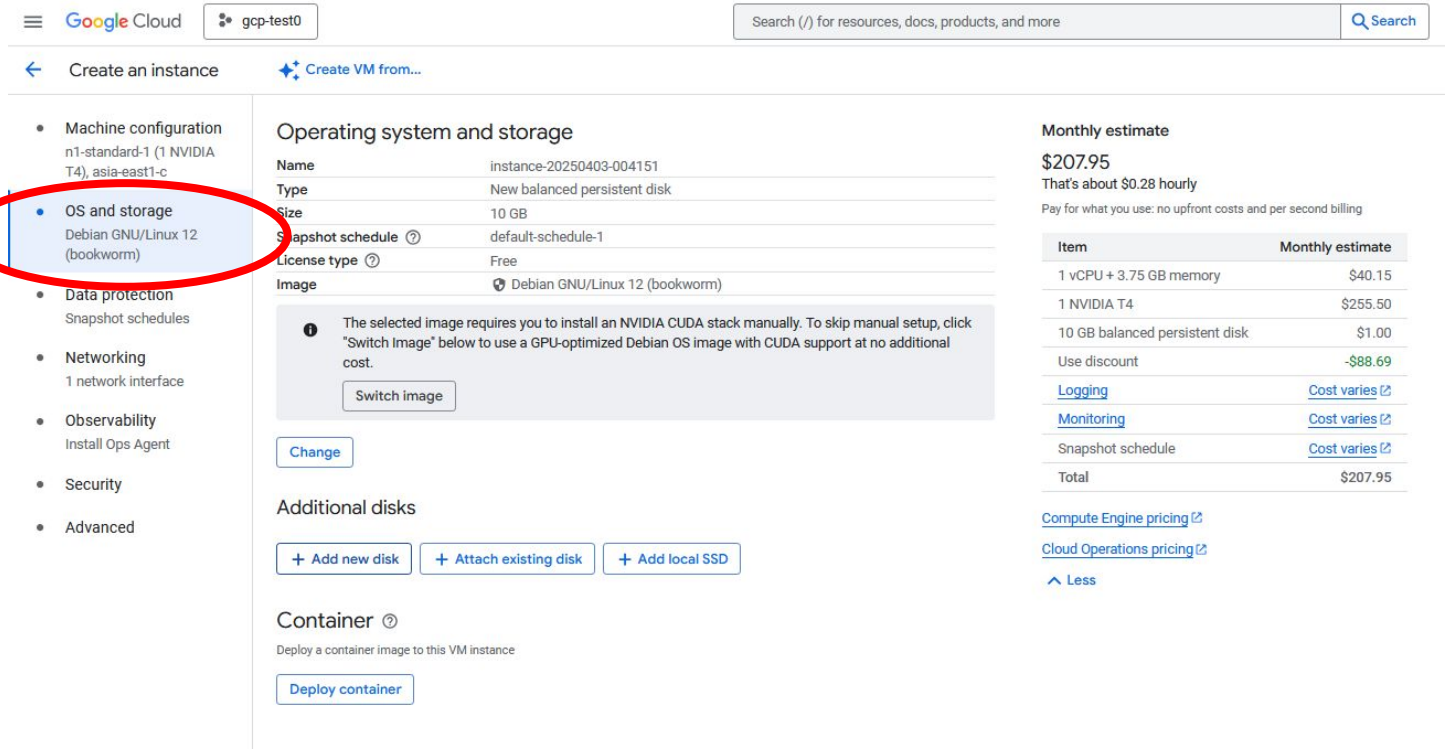
Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
1 vCPU + 3.75 GB memory	$39.05
1 NVIDIA T4	$270.10
10 GB balanced persistent disk	$1.10
Use discount	-$92.75
Total	$217.51

Compute Engine pricing

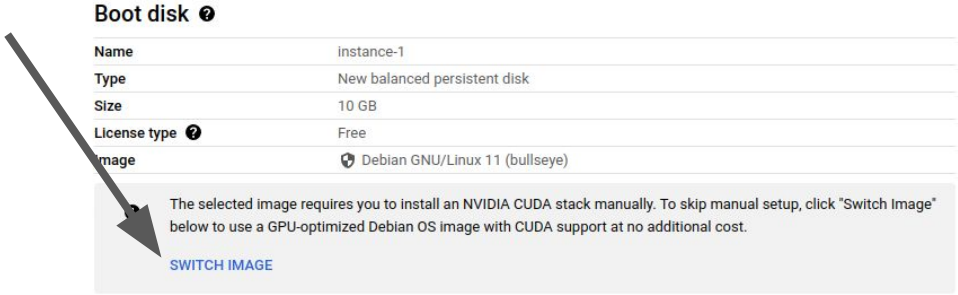
LESS

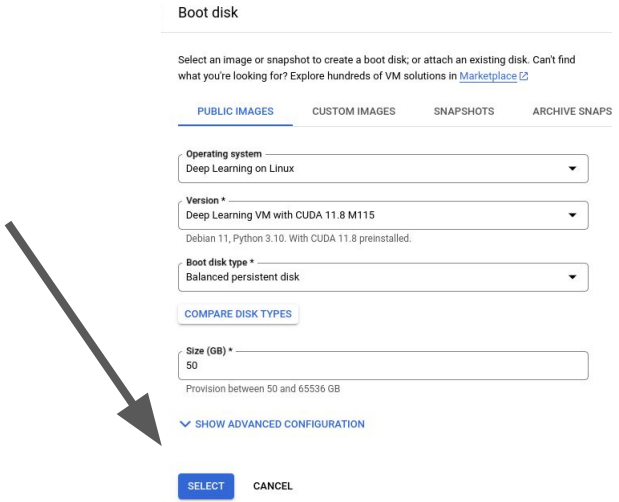
4. Create an instance with attached GPUs



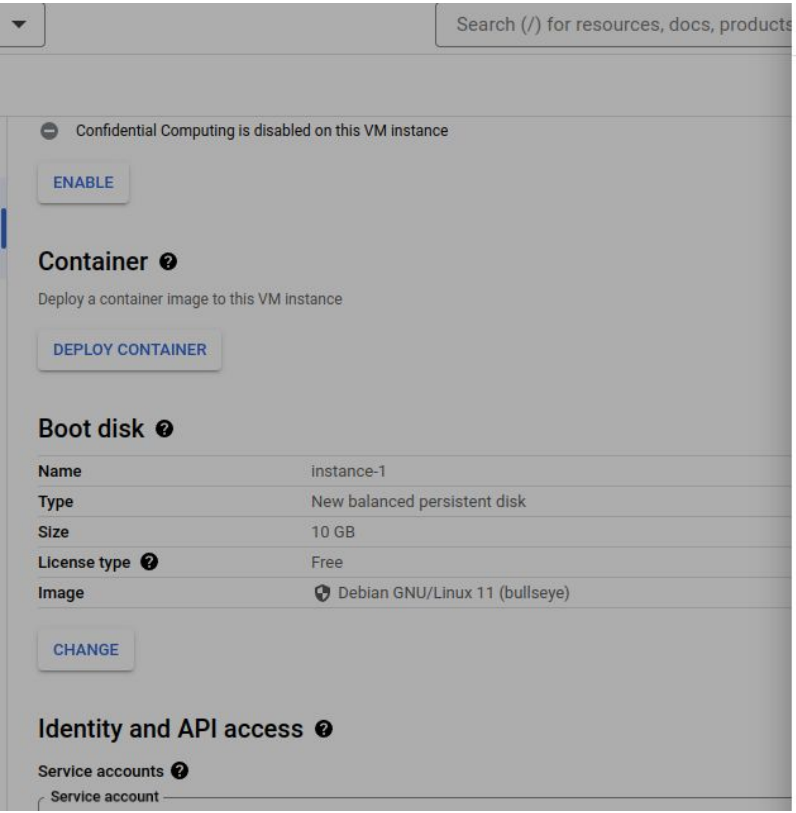
4. Create an instance with attached GPUs

- Choose Boot disk and image





- Change firewall setting
  - Select allow HTTP and HTTPS traffic, so you can install packages and connect to GitHub server
- Click ”Create”



Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in Marketplace

PUBLIC IMAGES | CUSTOM IMAGES | SNAPSHOTS | ARCHIVE SNAPSHOTS | EXISTING DISKS

Operating system
Deep Learning on Linux

Version *
Deep Learning VM with CUDA 11.8 M115

Debian 11, Python 3.10. With CUDA 11.8 preinstalled.

Boot disk type *
Balanced persistent disk

COMPARE DISK TYPES

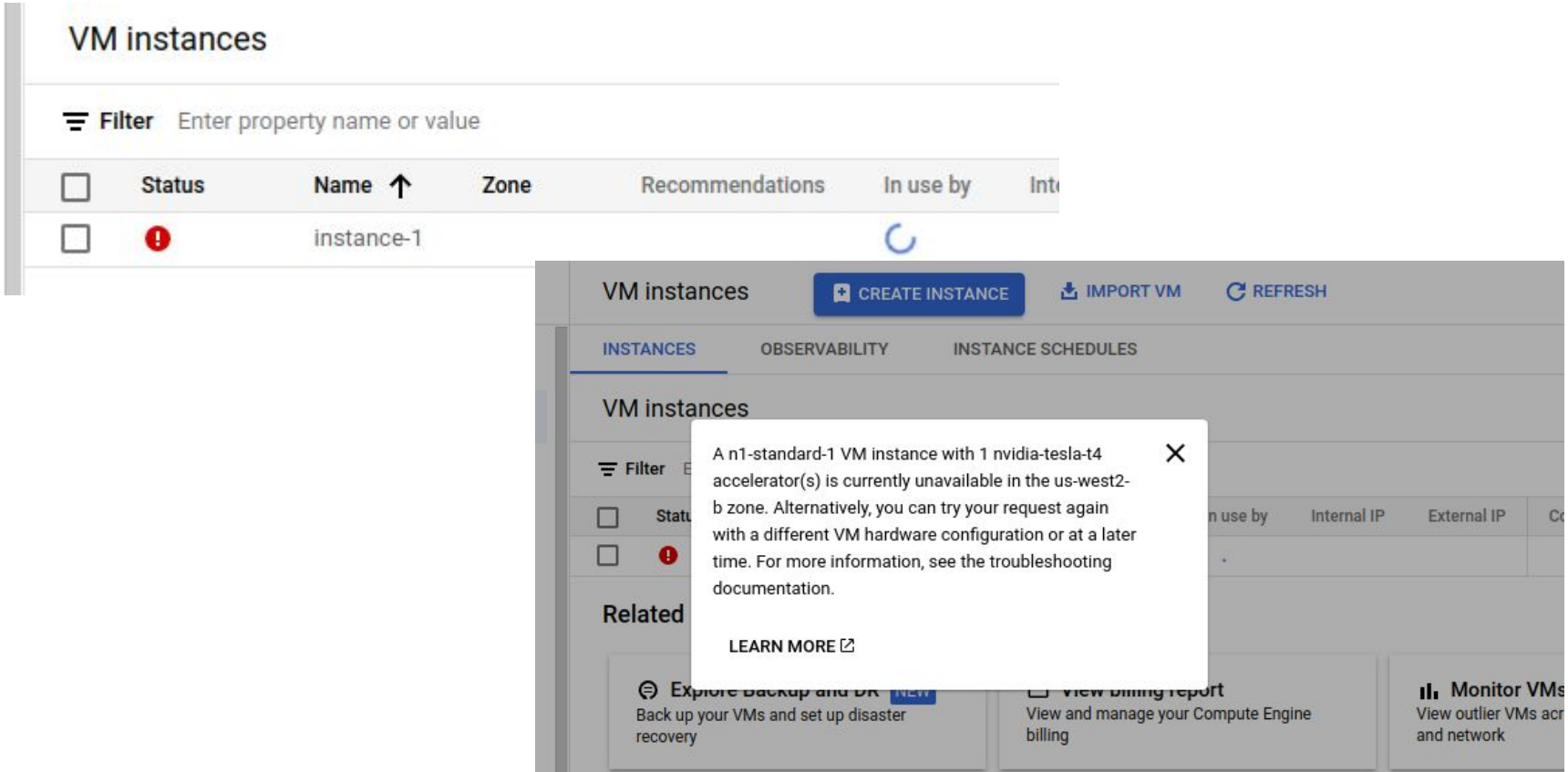
Size (GB) *
50

Provision between 50 and 65536 GB

SHOW ADVANCED CONFIGURATION

SELECT CANCEL

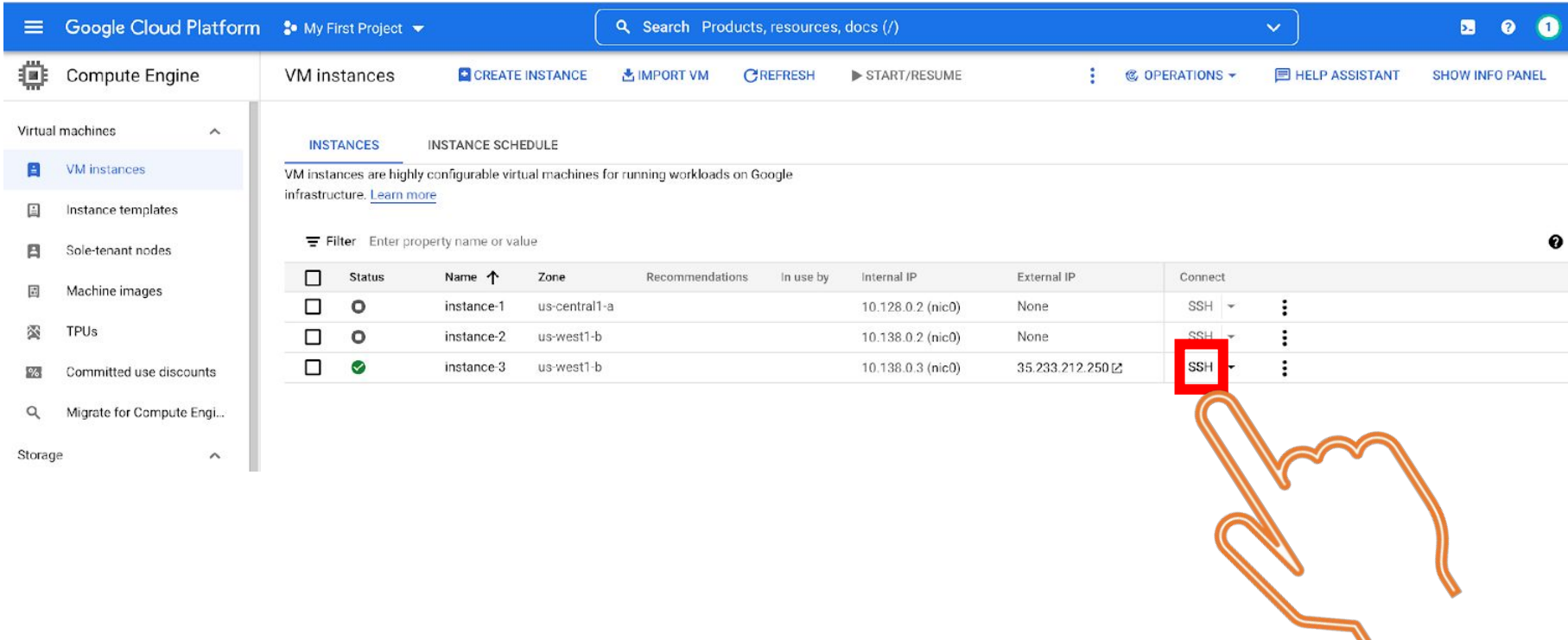
Note: If you have difficulty obtaining a machine, try using another region instead



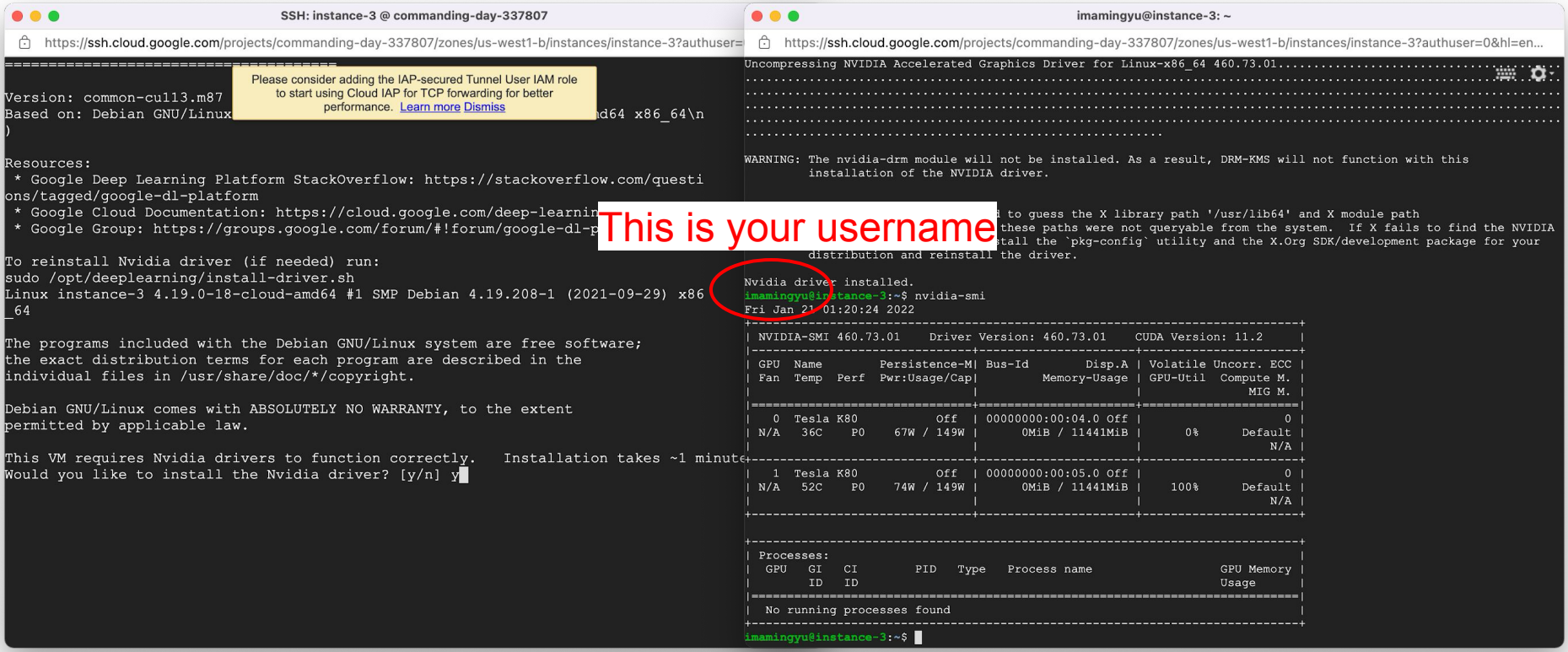
5. Install GPU driver

- If you choose the image with CUDA, your GPU driver will be installed automatically when you first login your machine
  - SSH into your machine in the Google Cloud portal (you have to login using your admin account to install the driver)
  - Input “y” when it prompts “Would you like to install the NVIDIA driver? ”

5. Install GPU driver



5. Install GPU driver



5. Install GPU driver

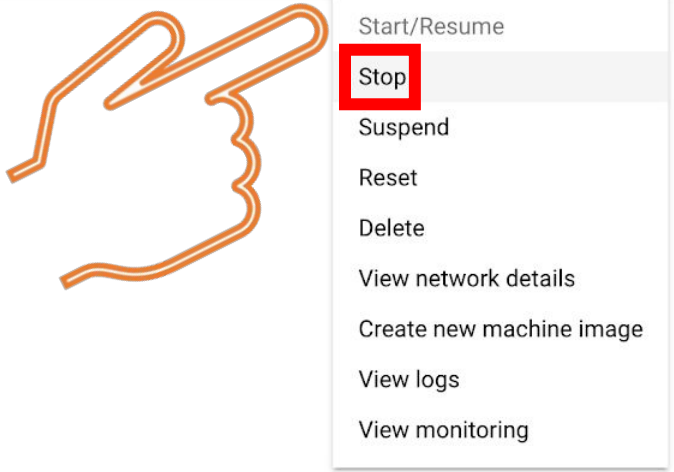
- Verify the GPU driver is installed
  - Type “nvidia-smi” command, you should see this if the driver is installed successfully

5. Install GPU driver

- Otherwise you could following steps in the following link
  - Installing GPU drivers | Compute Engine Documentation | Google Cloud
- We need to install
  - NVIDIA driver
  - CUDA toolkit
  - CUDA runtime

Turn off your machine when it’s not using

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
	instance-1	us-central1-a			10.128.0.2 (nic0)	None	SSH
	instance-2	us-west1-b			10.138.0.2 (nic0)	None	SSH
	instance-3	us-west1-b			10.138.0.3 (nic0)	35.233.212.250	SSH



- So you can save some credit

~~Setup Google Colab using VM as Backend~~

Google Colab is now deprecated from VMs.
More Info

Alternatives:

- Connect to VM with SSH and VSCode (recommended)
  - Develop code using a local VS Code editor | Cloud Workstations | Google Cloud
  - Remote Development using SSH
- gcloud CLI and JupyterLab
  - Quickstart: Create a Deep Learning VM instance by using the gcloud CLI
  - Connect to JupyterLab | Deep Learning VM Images | Google Cloud

Tips for Experiments on Remote Machine

Connect to your instance

- If you would like to connect to your machine using terminal directly, instead of using the browser-based ssh window
- Create key (Detailed tutorial: How to Use SSH Public Key Authentication – ServerPilot)
  - Using command ssh-keygen
  - You will keep the private key (for example id_rsa) in your local computer
- Add key
  - Add public key (like id_rsa.pub) to your Google Cloud instance setting
  - Click into your instance, click “Edit” in the top navigation bar, find “SSH key”, click “Add Item”, enter your public SSH key content there
- Connect your remote instance from your local terminal

  ssh –i key_path username@external_ip_address
- Connecting to Linux VMs using advanced methods | Compute Engine Documentation | Google Cloud

Access file and coding remotely

- You will need to edit code and run the updated codebase with new implementation
- Choice 1: VS Code
  - Developing on Remote Machines using SSH and Visual Studio Code
- Choice 2: PyCharm
  - Getting started with remote development | PyCharm (jetbrains.com)
- Choice 3: transfer files by scp/sftp
  - Using scp/sftp to transfer file/code from your local machine to the remote machine

Monitor and specify GPU usage

- Check whether your job is running on GPU, memory usage, job ID etc
  - nvidia-smi
- Specify which GPU(s) to use
  - export CUDA_VISIBLE_DEVICES="0”
  - export CUDA_VISIBLE_DEVICES="0,1,2”
  - export CUDA_VISIBLE_DEVICES=""

Run experiments in background

- Use tmux to run your job in background, so your job can continue running if your ssh session broke
- tmux new -s exp1
  - Create a new tmux session
- control + b, then press d
  - Exit the session
- tmux a –t exp1
  - Enter the session exp1 again
- tmux ls
  - See all active sessions

Install environment and run code

Install miniconda

>>> wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

>>> sh Miniconda3-latest-Linux-x86_64.sh

Create conda environment

>>> conda create -n ece239as python==3.xx

>>> conda activate ece239as

Install dependencies needed

>>> conda install pip

>>> pip3 --no-cache-dir install torch==1.10.1+cu113 torchvision==0.11.2+cu113 torchaudio==0.10.1+cu113 -f https://download.pytorch.org/whl/cu113/torch_stable.html