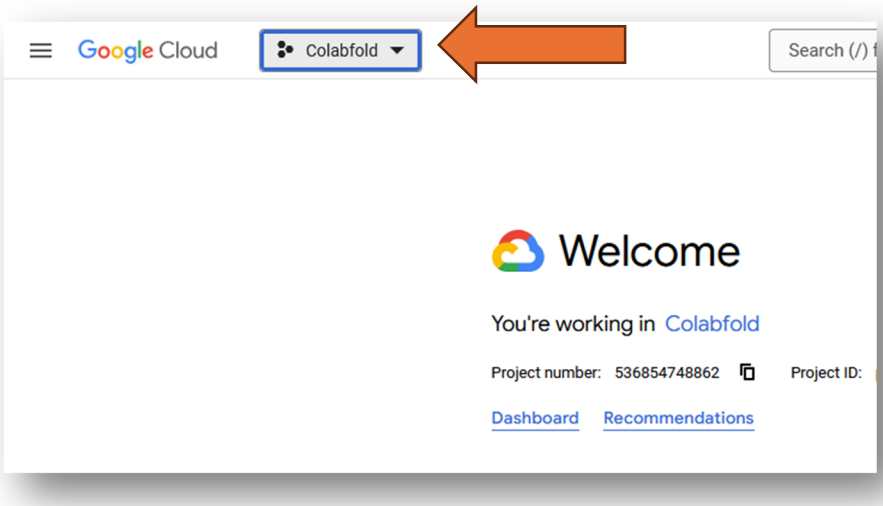
Running a Virtual Machine on Google Cloud Platform

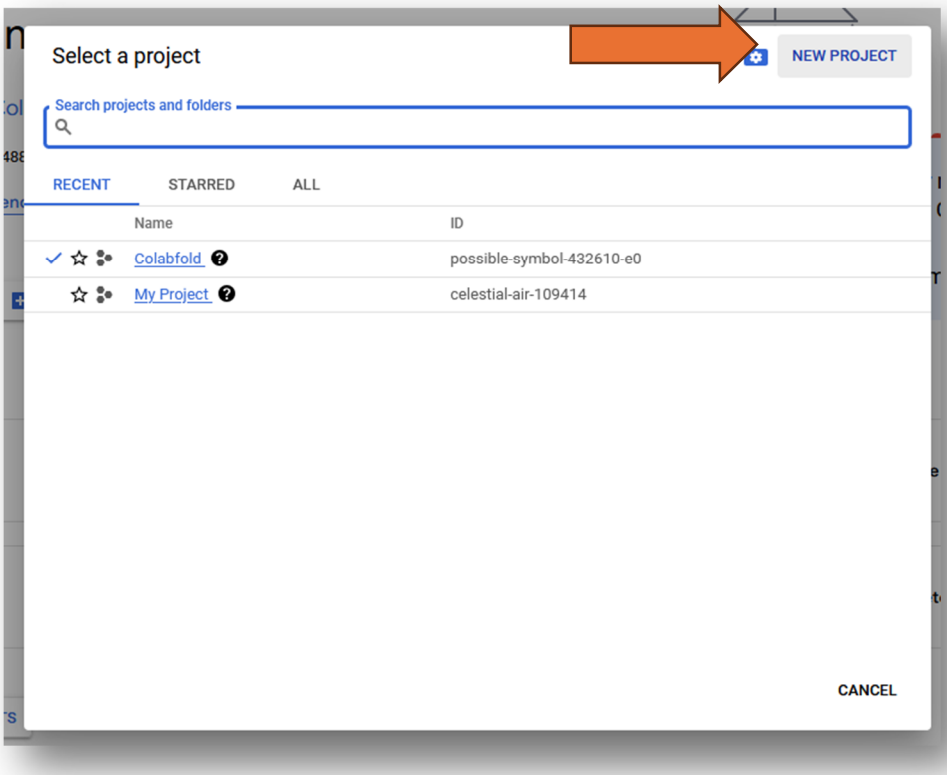
Set up a project

Go to: https://console.cloud.google.com

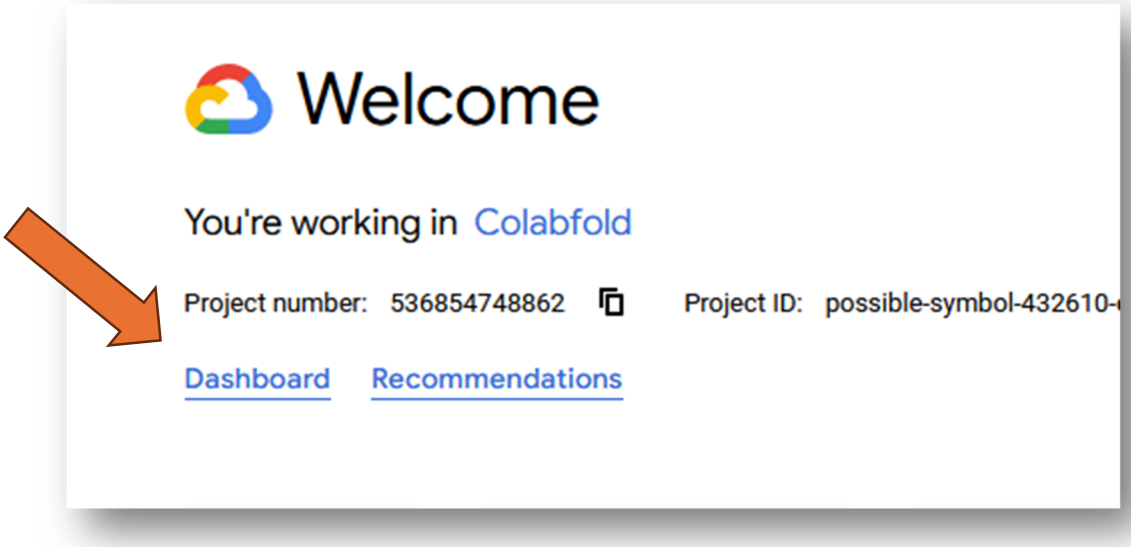
Click on the project dropdown



Click “New Project”



Click “Dashboard”



Get billing access

GCP will give you some free credits if you’re a new user, though you’re limited to what these can be spent on. If you want to use other services, such as using GPU’s, you’ll need to have access to a billing account.

Read how to do this here: https://cloud.google.com/billing/docs/how-to/billing-access

Check quotas

To be able to request resources, they must be within your quota assigned by GCP.

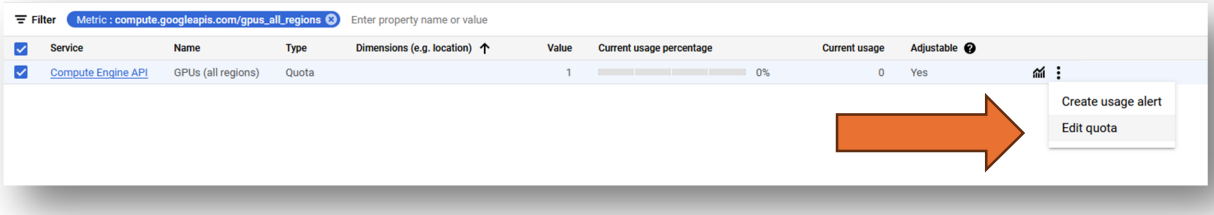
From any page “/” allows you to search, this seems to be the easiest way to navigate.

Go to the “Quotas & System Limits” page.

The default GPU quota is 0, if you want to use a GPU, you will need to increase this quota.

To increase the GPU quota, in the quota search field enter “gpus_all_regions”

Click the menu button for the service, then “Edit quota”



On submitting the request to Google, you will likely receive an automated email telling you the request has been made, and shortly after a second email detailing what your quota has been set as. I requested five GPUs and was allocated one.

Create a Virtual Machine

Resources are hosted in different geographical locations known as “regions”. Various factors determine which region you may decide to use. Some factors to consider when choosing a region:

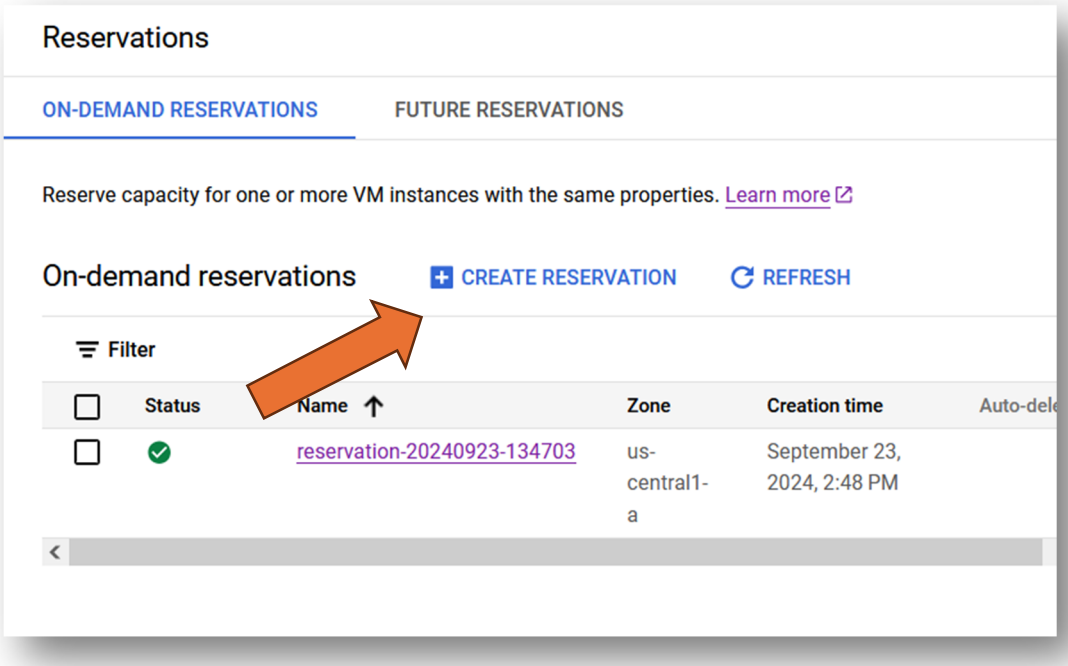
- Not all resources are available in all regions
- The same resource may be priced differently in different regions
- Latency (if you are using other microservices, you may experience latency if they are in a different regions)
- Data governance and privacy concerns
- CO_2 emissions

Here’s some resources to help you decide:

- Documentation (https://cloud.google.com/compute/docs/regions-zones/)
- Region picker (https://cloud.withgoogle.com/region-picker/)
- GPU availability (https://cloud.google.com/compute/docs/gpus/gpu-regions-zones)
- Virtual Machine pricing (https://cloud.google.com/compute/all-pricing)

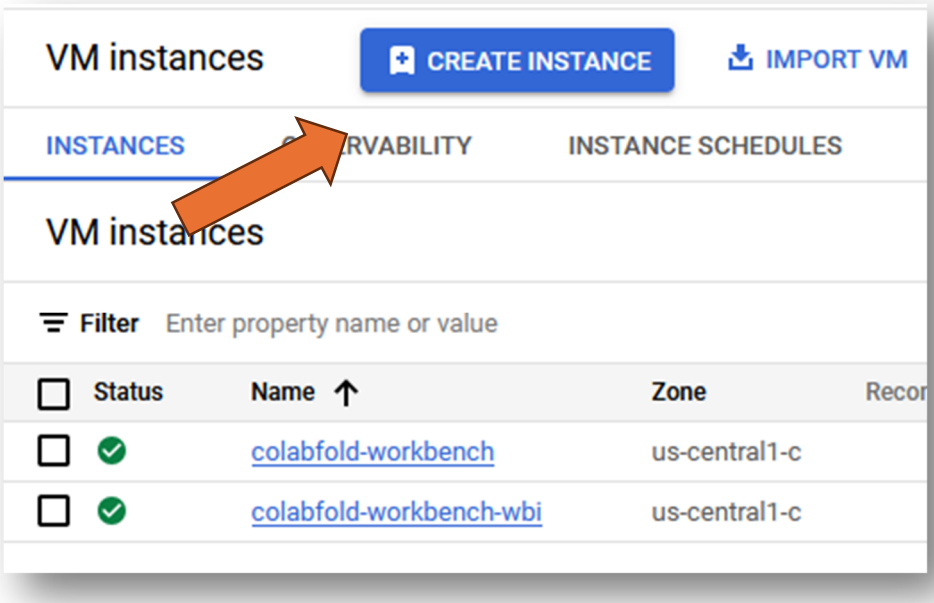
Even if the resources you are requesting are within quota, you may not be able to use them due to demand. Reserving resources gives you a “very high level of assurance” that you will be able to obtain them.

To reserve a VM, go to the “Reservations” page under “Compute Engine”, and click “Create Reservation”. Fill out the form.



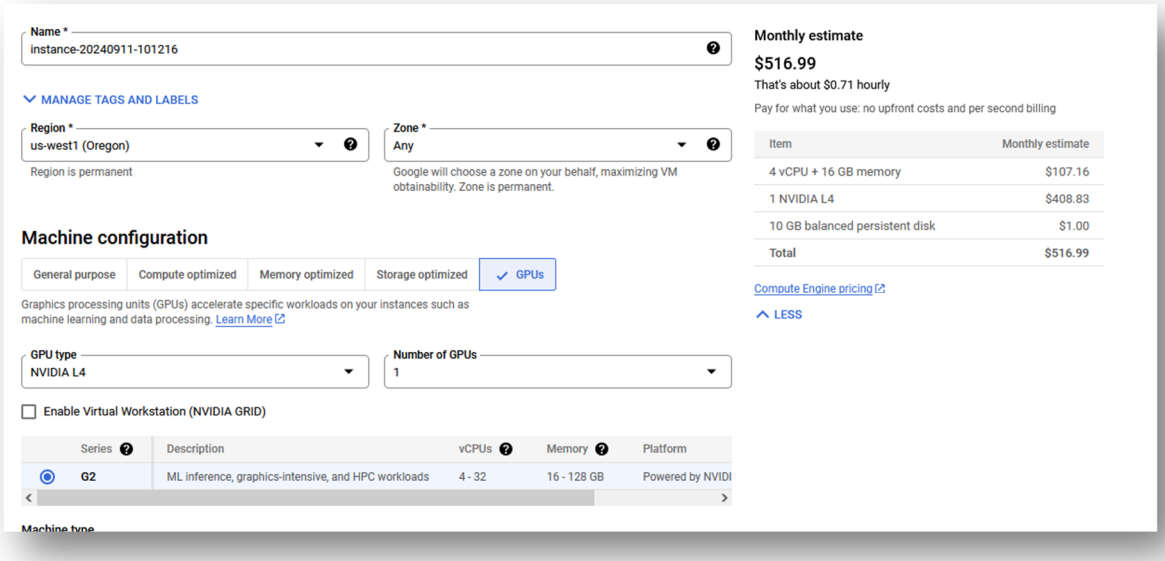
Note that as soon as the reservation is created, you will start being charged for the use of those resources.

To create a VM, whether it has been reserved or not, go to the “Compute Engine” page and click “Create Instance”



Select the configuration that you require, GCP will give you a monthly cost estimate based on your selection.

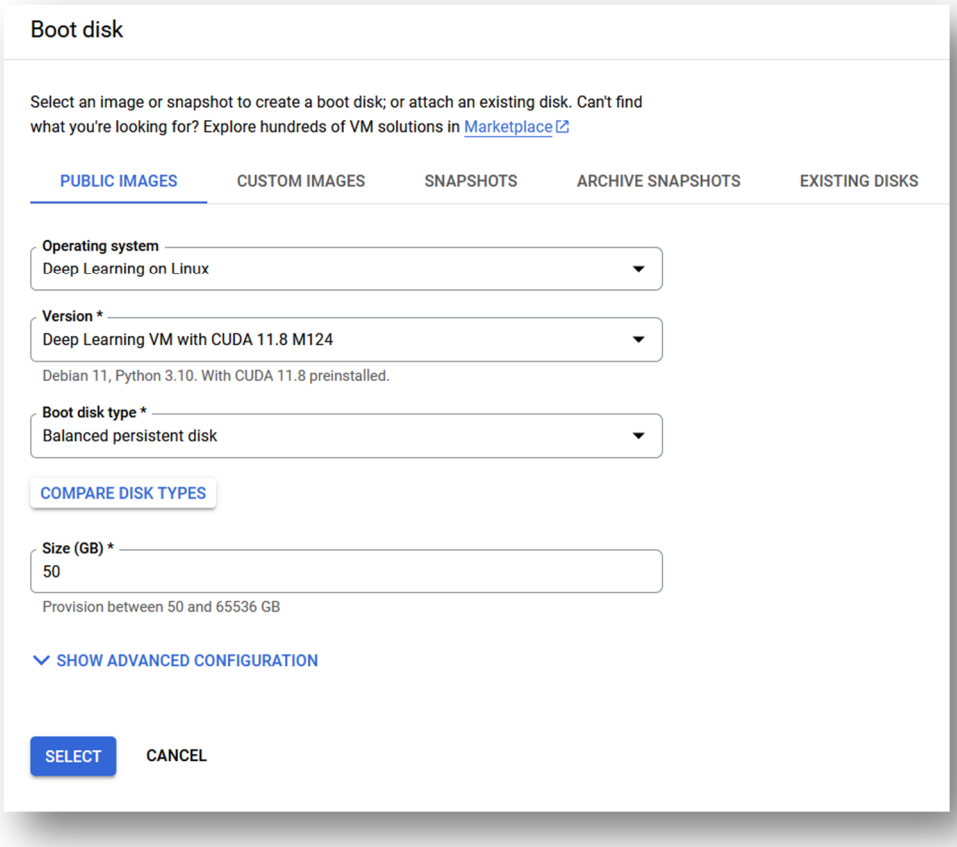
If you wish to use a reservation, create an instance that matches the zone, machine type, CPU platform, GPU amount and type, and local SSD interface and size. Also make sure in “Advanced options” -> “Management” -> “Reservations” that “Automatically use created reservations” is selected.



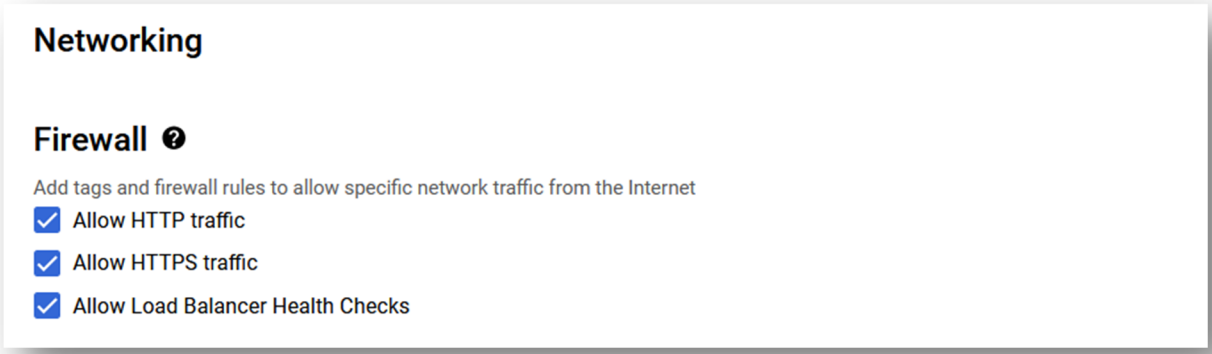
You will have to increase your quota if you wish to request more resources than your quota currently allows, e.g. for GPU, VM storage etc.

Choose the OS that best suits your needs, note that if you wish to use GPUs you can select to have CUDA preinstalled.

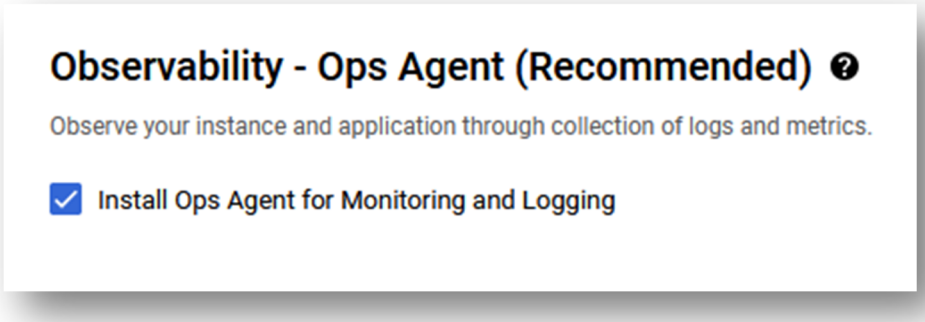
Select how much storage capacity you want for your VM. This is easier than using GCP's microservices for storage.



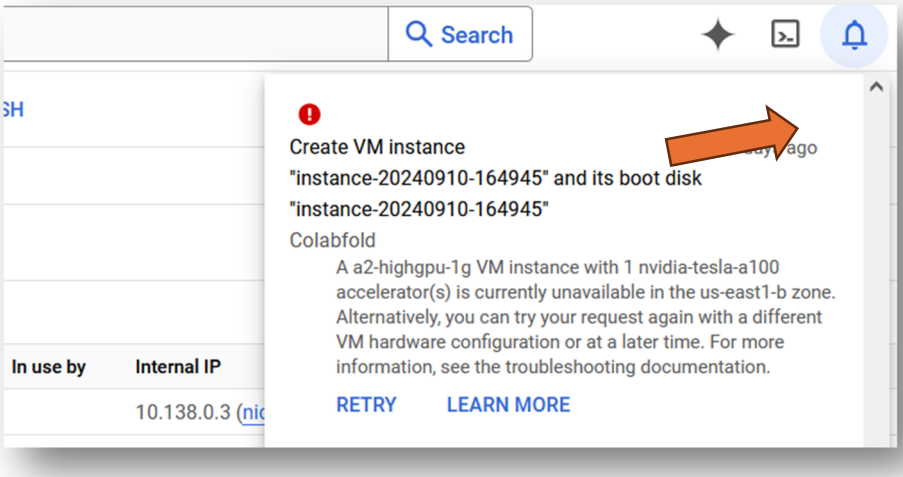
By default all incoming traffic is blocked, so you may wish to allow the appropriate traffic if you wish to access the internet.



It's recommended to tick "Install Ops Agent for Monitoring and Logging" so that you can monitor the performance of your VM.



If resources are unavailable when you request them, when you click "Notifications" you'll get an error message similar to below. This should not happen if you create a reservation.



Access via VSCode

Below are steps to access your VM via VSCode, though similar steps should work for other text editors/ IDE's.

Install the gcloud CLI (https://cloud.google.com/sdk/docs/install)

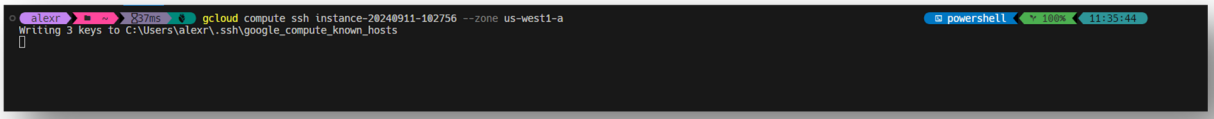
Run "gcloud compute ssh (INSTANCE_NAME) --zone (ZONE)"

Where instance name and zone are given on the "VM instances" page (from the Google Cloud console you can search using "/ VM instances")

VM instances page

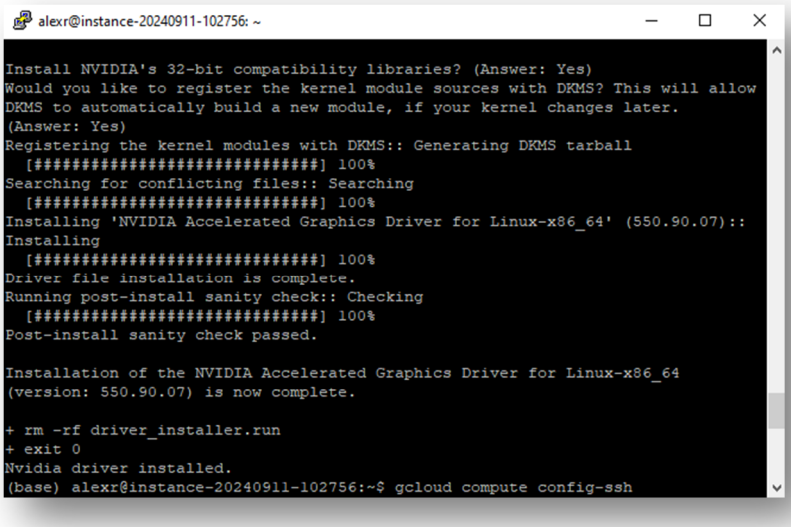
So in this instance: “gcloud compute ssh instance-20240911-102756 --zone us-west1-a”

VSCode terminal



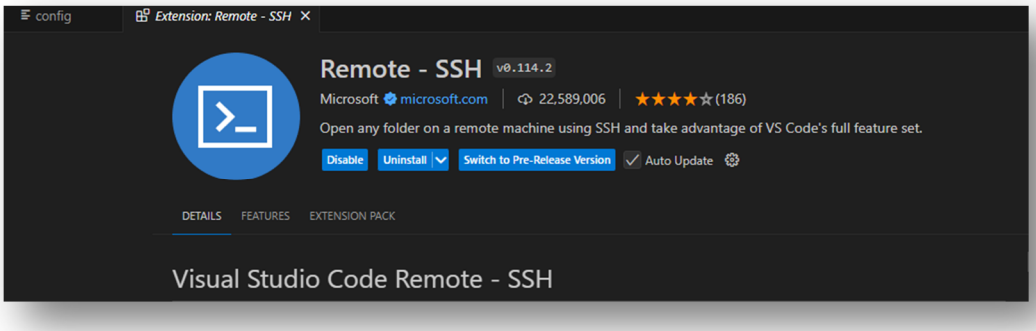
When first SSHing into the machine, a new putty window will open and some setup will take place, including installing CUDA if requested.

Once this process has run, in the putty window, configure the SSH configuration file to make SSHing easier by running “gcloud compute config-ssh”. Note, you may need to run “gcloud auth login” and follow the instructions if you get an error regarding “Insufficient Permission”

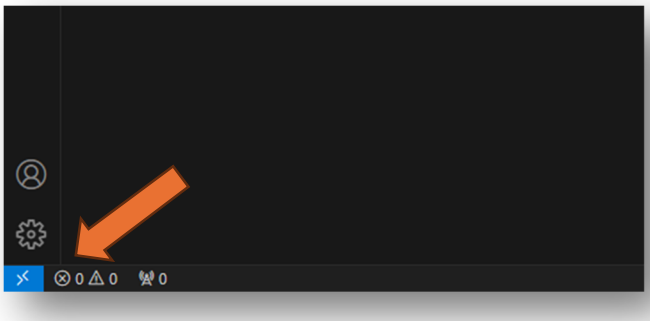


You can then close the putty window.

In VSCode, install the remote-ssh extension if it isn’t already installed.



In the bottom left of the VSCode window, click the “Open a Remote Window” button.



In the dropdown select “Connect to Host…”, then “Configure SSH Hosts…”, then select your ssh config file. Enter the following details:

Host (a nickname or abbreviation for the host)

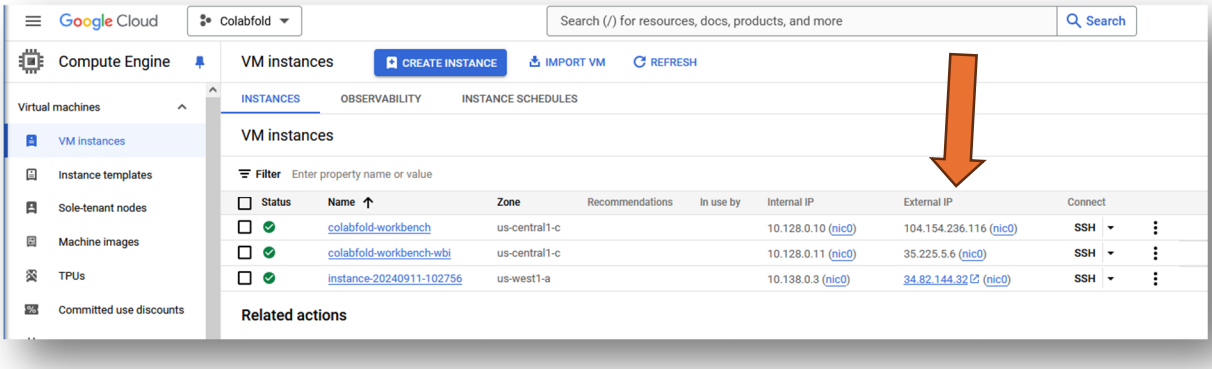
HostName (External IP)

AddKeysToAgent yes

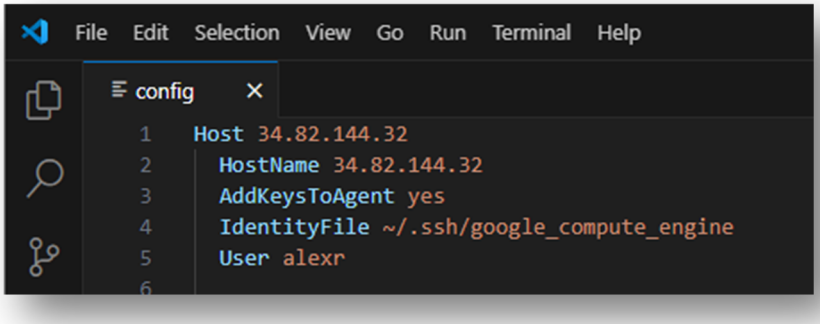
IdentityFile ~/.ssh/google_compute_engine

User (your local username)

Where “External IP” can be found on the VM instances page

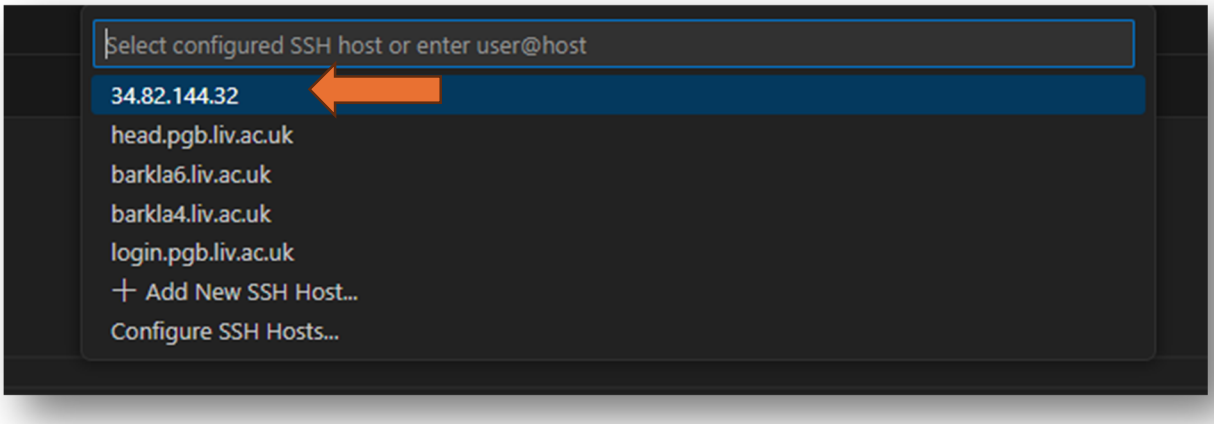


For example:

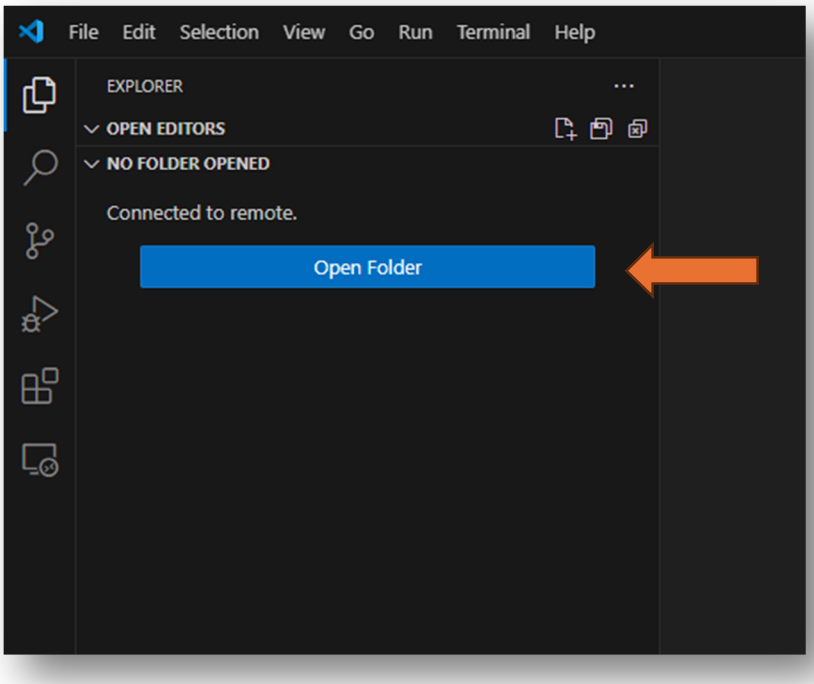


The VM should then be available as a host when you next click “Open a Remote Window” in VSCode.

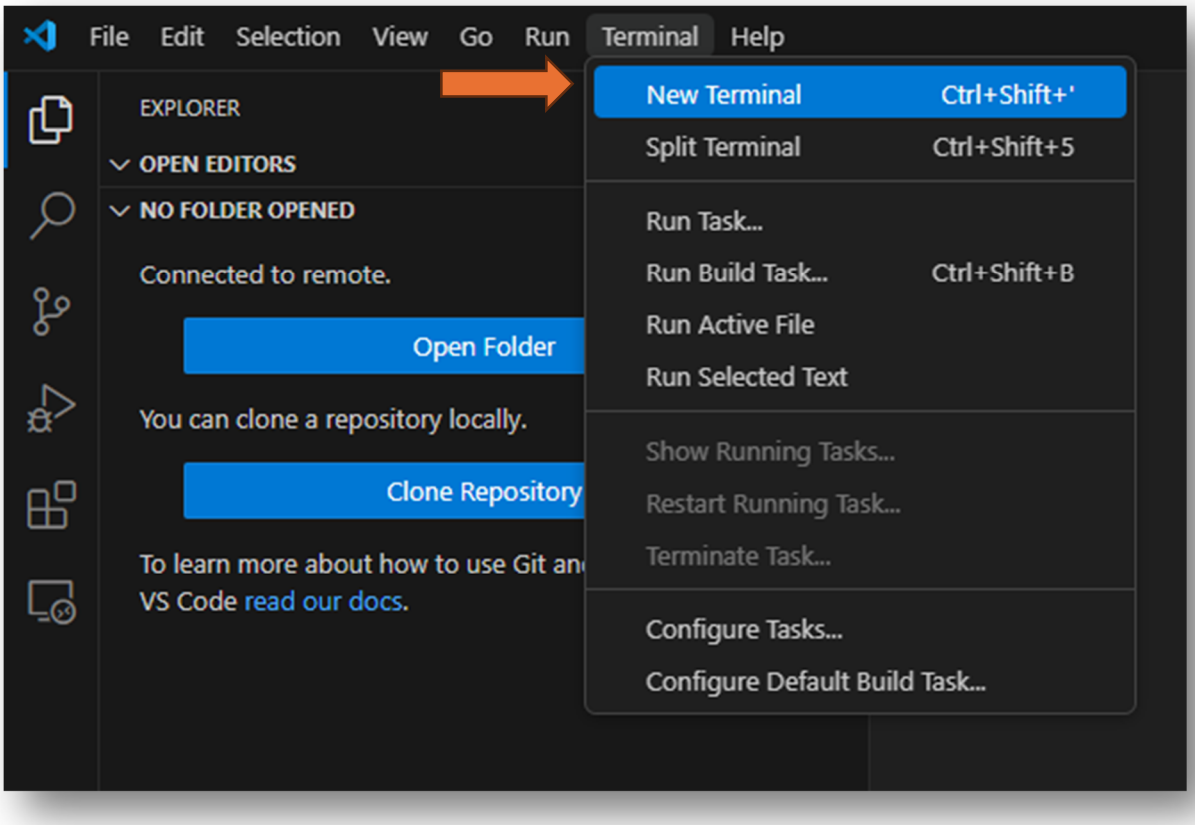
Click the External host name to connect to it:



On the left you can then click “Open Folder” and choose which folders you would like to browse.



Open a terminal window by going to “Terminal” → “New Terminal”



You should now have access to files and a terminal.

Access via WinSCP

VSCode does not have a good interface for transferring files.

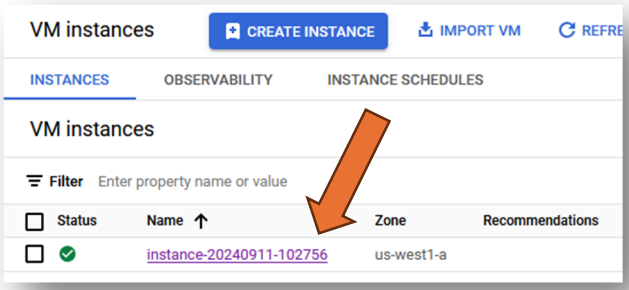
If you’re on Windows, you may want to use WinSCP. To access the VM via WinSCP follow these instructions: https://cloud.google.com/compute/docs/instances/transfer-files#winscp

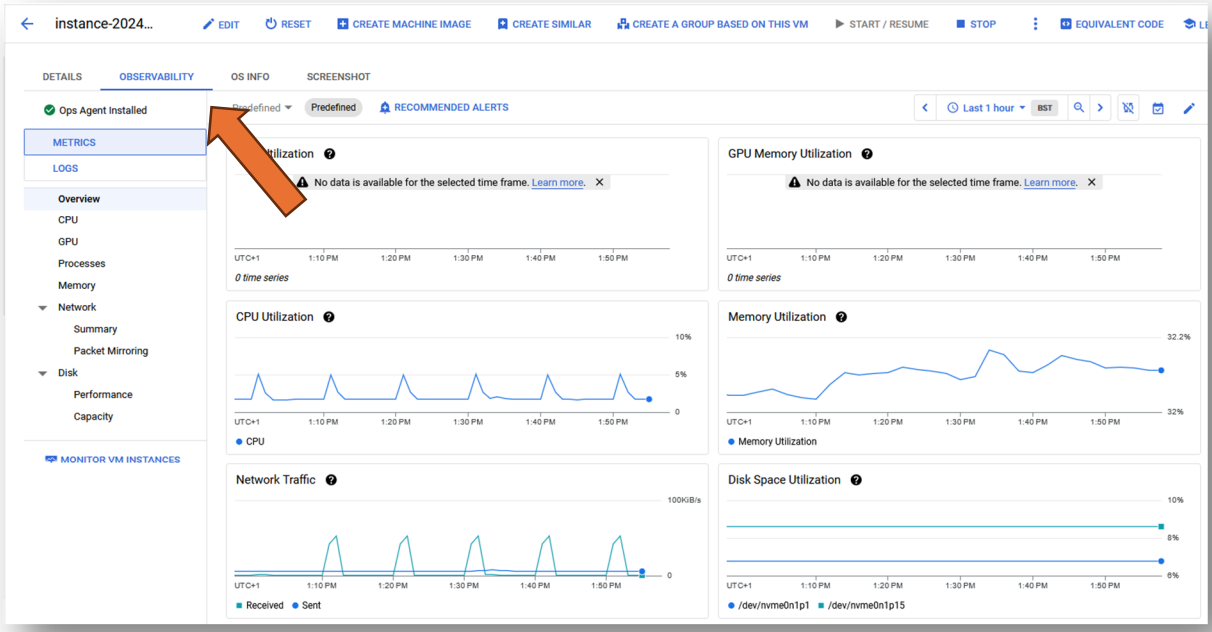
Note that by default on Windows, the private key for your compute engine is generated at: "C:\Users\(username)\.ssh\google_compute_engine.ppk"

If you’re on a Mac or Linux, see here: https://cloud.google.com/compute/docs/instances/transfer-files#scp

Resource monitoring

Resources can be monitored if Ops Agent has been installed when creating the instance. This can be accessed by going to the “Monitoring” page, or clicking the name of the instance on the “VM instances” page, then going to the “Observability” tab.





Tmux

If you close your SSH connection, then by default any process you are running will also end.

tmux allows you to detach a process to keep it running in the background.

To create a new tmux session use “tmux new”, you can then start your process in this session.

To detach from this session and let the process continue running in the background: press “ctrl + b, d”.

To reattach to the session “tmux attach”.

GPU scheduling

If you've requested a VM with multiple GPUs and want to run different processes on each GPU, "simple-gpu-scheduler" can make this easier.

It can be installed using "pip install simple-gpu-scheduler" (https://pypi.org/project/simple-gpu-scheduler/).

Example usage to run a batch of fastas through colabfold on all available gpu's:

```
# count all gpu's available
gpu_count=$(nvidia-smi -L | wc -l)

# write commands to be run to txt file for scheduler
for f in "${fasta_names[@]}";
do
    echo "colabfold_batch --num-models 3 --use-gpu-relax --amber ${fasta_path}${f} colabfold_output/" >> colabfold_commands.txt
done

# run using gpu scheduler
simple_gpu_scheduler --gpus $(seq -s ',' 0 $(($gpu_count-1))) < colabfold_commands.txt
```

Ending the VM

To keep an eye on costs, check the “Billing” page and set up budget alerts.

Before you close the VM, make sure you’ve downloaded any data stored on the VM.

From the “VM Instances” page, click the menu button associated with the instance, then click “Stop” to end the VM, or “Suspend” to pause the VM. Note that when suspended, the VM doesn’t incur any expenses other than those used to store the VMs memory.