CSCI-GA 3033

Sec-085

Cloud and Machine Learning

Homework-2
GPU Chase

Under the guidance of
Prof. Hao Yu and Prof. I-Hsin Chung

Done by
Prithvi Chilka (pc3427)
(UID): N14739560

GPU Chase: Find a GPU in Google Cloud

1 Introduction

1.1 Background

In a competitive cloud computing environment, small companies struggle to access GPUs for AI tasks due to high demand and limited availability, particularly without premium support from providers like Google Cloud Platform (GCP).

1.2 Purpose

The goal is to automate the process of locating and allocating a GPU within GCP across different regions and zones, enabling efficient resource acquisition for urgent AI model training.

1.3 Approach

The strategy involves developing an automated script, potentially using Python or shell commands, that systematically searches through GCP's regions and zones for available GPUs. This script will then attempt to create a VM with the identified GPU, tracking success or failure reasons. Highlighting GPU availability and allocation success across tested zones.

2 Methodology

My script automates the search for available GPU resources across all operational zones within a Google Cloud Platform (GCP) project, aiming to create a virtual machine (VM) with an attached GPU for urgent AI model processing tasks. Here's a detailed breakdown of it:

2.1 Initialization

The script starts by defining essential variables such as the GCP project ID, the base image for the VM (Debian 12), and the boot disk size (200GB). These settings ensure that the script targets the correct project and creates VMs with a consistent environment and sufficient storage.

2.2 GPU Discovery

It employs a function, **get_first_available_gpu**, to list GPU types available in a specific zone that match the "nvidia*" name pattern. This function returns the first available GPU type, optimizing the search process by focusing on NVIDIA GPUs, widely used for AI and machine learning tasks.

2.3 Zone Iteration and VM Creation

The script iterates through each operational zone within the GCP, using **gcloud compute zones list** to dynamically retrieve a list of zones. For each zone, it checks for available GPU types using the previously defined function. Upon finding an available GPU, it determines the appropriate machine type based on the GPU's model, ensuring compatibility and optimal performance.

2.4 VM Creation Attempt

For zones with available GPUs, the script attempts to create a VM named **gpu-vm-<zone>**, configuring it with the selected GPU type, machine type, image family, and other specified settings. A startup script is included to install the CUDA Toolkit (1) on the VM, facilitating AI and machine learning operations.

2.5 Storing Results in Tabular Format

The script incorporates a function, **store_output**, to meticulously format and append each GPU search and VM creation attempt into a structured text file, **gpu_availability.txt**. This output file is initialized with headers indicating Zone, GPU Type, VM Name, Creation Status and Comments. After each attempt, whether successful or failed, the script captures and records the details into this table, providing a clear and concise overview of the availability and allocation outcomes for quick reference and analysis.
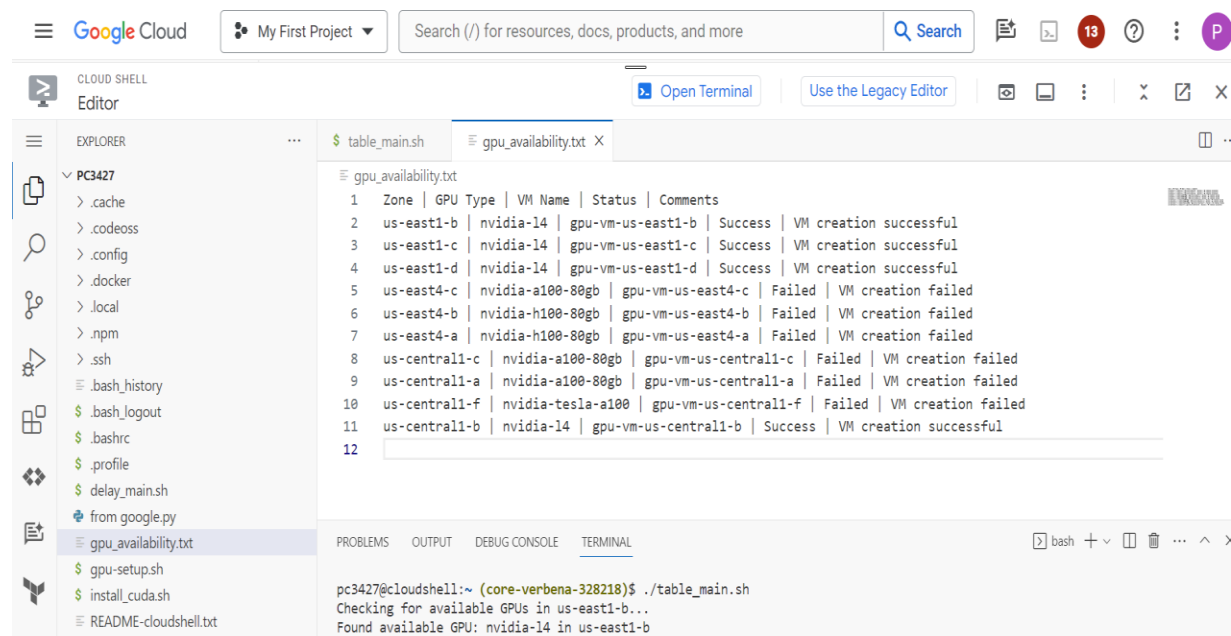
2.6 Verification and Cleanup

After VM creation, the script waits for the VM to initialize, then verifies the GPU's functionality by connecting via SSH and running commands to detect the NVIDIA GPU. This step confirms the successful setup of the GPU environment. Following verification, the script cleans up by deleting the VM, preventing unnecessary costs.

3 Summary

The output of the GPU Chase script is efficiently summarized in a table within **gpu_availability.txt**. This table records each attempt to secure a GPU across various zones, detailing the type of GPU, the assigned VM name, the status of the VM creation, and any pertinent comments. Successful entries indicate a VM was created without issues. Failures are noted, particularly for the nvidia-a100 and nvidia-h100 GPUs, **due to a lack of quota** within the organization for these types. This comprehensive tabulation allows for at-a-glance assessment of GPU availability and allocation success, streamlining resource management efforts.

4 Output

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                                          >_ bash  + ∨  ⊞  🗑  ⋯  ∨  ✕

Checking for available GPUs in us-east1-d...
Found available GPU: nvidia-l4 in us-east1-d
Attempting to create VM (gpu-vm-us-east1-d) in zone us-east1-d with GPU type nvidia-l4 and machine type g2-standard-4
Created [https://www.googleapis.com/compute/v1/projects/core-verbena-328218/zones/us-east1-d/instances/gpu-vm-us-east1-d].
WARNING: Some requests generated warnings:
 - Disk size: '200 GB' is larger than image size: '10 GB'. You might need to resize the root repartition manually if the operating system does
not support automatic resizing. See https://cloud.google.com/compute/docs/disks/add-persistent-disk#resize_pd for details.

NAME: gpu-vm-us-east1-d
ZONE: us-east1-d
MACHINE_TYPE: g2-standard-4
PREEMPTIBLE:
INTERNAL_IP: 10.142.0.16
EXTERNAL_IP: 34.74.70.62
STATUS: RUNNING
VM (gpu-vm-us-east1-d) creation successful in zone us-east1-d with GPU nvidia-l4
Waiting for VM (gpu-vm-us-east1-d) to initialize...
Verifying GPU installation on VM (gpu-vm-us-east1-d)...
Warning: Permanently added 'compute.65446242352008650' (ECDSA) to the list of known hosts.
00:03.0 3D controller: NVIDIA Corporation AD104GL [L4] (rev a1)
GPU verified successfully on VM (gpu-vm-us-east1-d)
Cleaning up: Deleting VM (gpu-vm-us-east1-d) to avoid unnecessary costs.
Deleted [https://www.googleapis.com/compute/v1/projects/core-verbena-328218/zones/us-east1-d/instances/gpu-vm-us-east1-d].
Checking for available GPUs in us-east4-c...
Found available GPU: nvidia-a100-80gb in us-east4-c
Attempting to create VM (gpu-vm-us-east4-c) in zone us-east4-c with GPU type nvidia-a100-80gb and machine type a2-ultragpu-1g
WARNING: Some requests generated warnings:
 - Disk size: '200 GB' is larger than image size: '10 GB'. You might need to resize the root repartition manually if the operating system does
not support automatic resizing. See https://cloud.google.com/compute/docs/disks/add-persistent-disk#resize_pd for details.

ERROR: (gcloud.compute.instances.create) Could not fetch resource:
 - Quota 'NVIDIA_A100_80GB_GPUS' exceeded.  Limit: 0.0 in region us-east4.
        metric name = compute.googleapis.com/nvidia_a100_80gb_gpus
        limit name = NVIDIA-A100-80GB-GPUS-per-project-region
        limit = 0.0

                                          Ln 12, Col 1   Spaces: 4   UTF-8   LF   Plain Text   ◆ Duet AI   ⌐
```

References

1. **Google.** Google Cloud GPU Drivers. *install GPU drivers.* [Online] Google, December 22, 2023.

2. **Medium.** towardsdatascience. *Installing cuda .* [Online] towardsdatascience, May 9, 2020.

3. **Google.** Cloud Tutorials. *Google cloud tutorials.* [Online] Google, September 29, 2023.

4. —. Google CLI . *Google CLI cheat sheet.* [Online] Google, January 20, 2024.