Elastic Computing Crash Course

CSCI 8360 Data Science Practicum

GCP and AWS

- From Google and Amazon, respectively

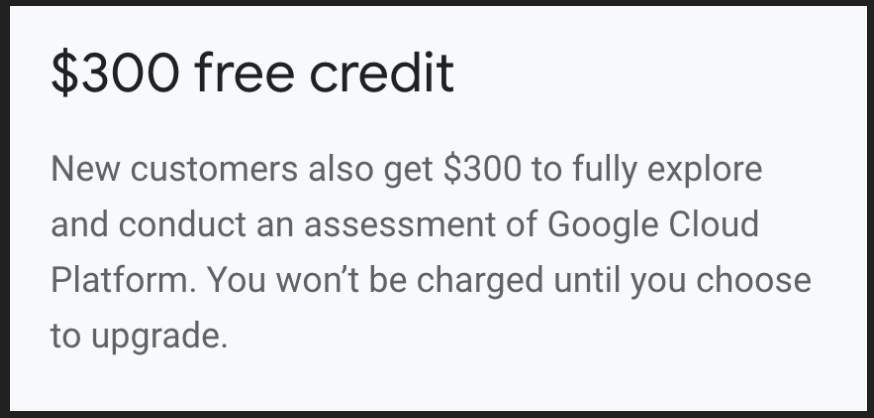
Google Cloud

Google Cloud Platform

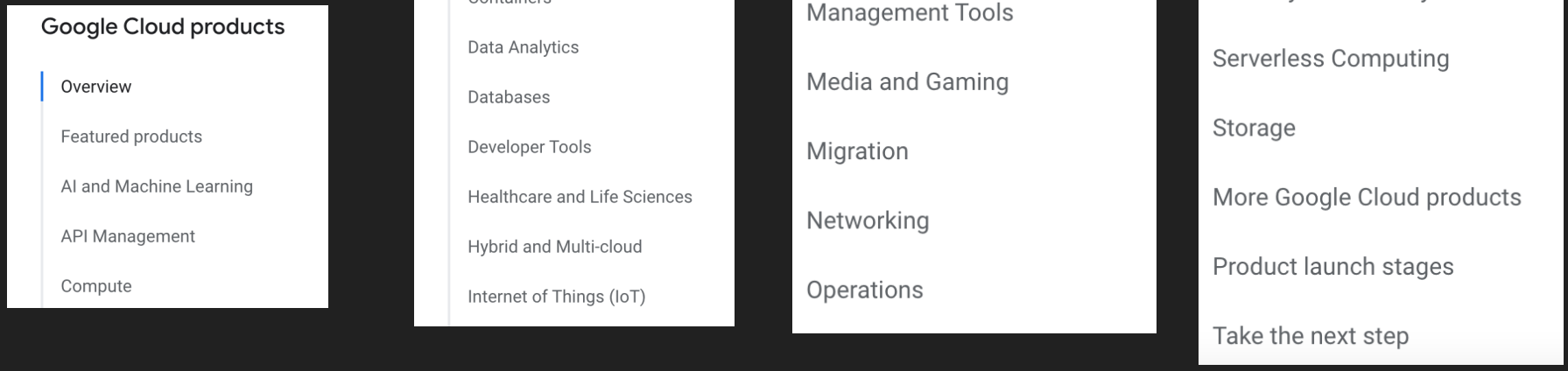
https://cloud.google.com/

GCP: New “customers”

- If you’ve never used GCP before, you are entitled to a one-time $300 credit!
- **Look into this before you redeem your class credits**

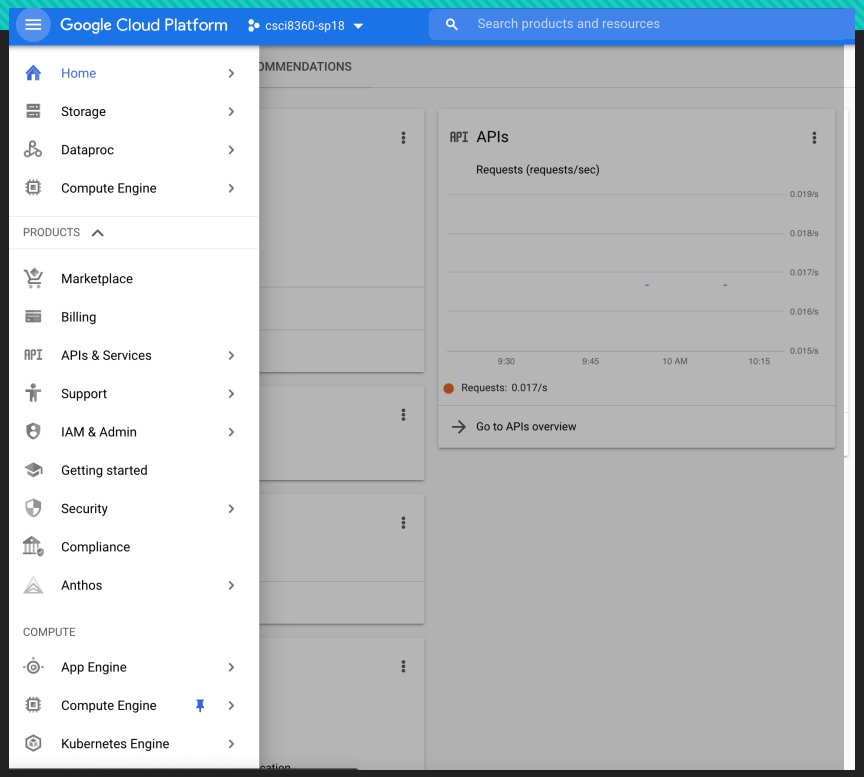


GCP Product Lineup



Google Dashboard

- Heads-up display of your active projects and their usage, billing, and associated resources
- Sidebar for *all* GCP products, quicklink-able



GCP Compute Engine

- This is your generic "virtual compute instances" product
- Different VM templates optimized for different tasks (and priced accordingly)

- General-purpose compute
- Compute-optimized (high CPU count)
- Memory-optimized (high RAM)
- Storage-optimized (large SSD/HDD)
- GPU instances

GCP Compute Engine

- Each compute instance has a (1) region, and an (2) hourly rate
- **Region** denotes the physical geographic location of the VM
  - Probably want to stick with east coast VMs; latency is better
- **Rate** denotes hourly cost of running the VM
  - 60 minutes and 1 second is billed as 2 hours
  - Billed as long as the VM is **on**; does not need to be doing anything! **Shut down your VMs when you are done to avoid extra charges**

GCP Compute Engine

E2 standard machine types

Iowa (us-central1) ▾ | Monthly ⬤ Hourly

Machine type	Virtual CPUs	Memory	Price (USD)	Preemptible price (USD)
e2-standard-2	2	8GB	$0.067006	$0.020102
e2-standard-4	4	16GB	$0.134012	$0.040204
e2-standard-8	8	32GB	$0.268024	$0.080408
e2-standard-16	16	64GB	$0.536048	$0.160816
e2-standard-32	32	128GB	$1.072096	$0.321632
Custom machine type	If your ideal machine shape is in between two predefined types, using a custom E2 machine type could save you as much as 40%. For more information, see E2 custom vCPUs and memory.			

GCP Compute Engine

- Note: GPUs are not cheap!
- Keep this in mind for projects when you're training deep learning models

NVIDIA® Tesla® V100	1 GPU	16 GB HBM2	$2.48 per GPU	$0.74 per GPU
	2 GPUs	32 GB HBM2		
	4 GPUs	64 GB HBM2		
	8 GPUs	128 GB HBM2		
NVIDIA® Tesla® P100	1 GPU	16 GB HBM2	$1.46 per GPU	$0.43 per GPU
	2 GPUs	32 GB HBM2		
	4 GPUs	64 GB HBM2		
NVIDIA® Tesla® K80	1 GPU	12 GB GDDR5	$0.45 per GPU	$0.135 per GPU
	2 GPUs	24 GB GDDR5		
	4 GPUs	48 GB GDDR5		
	8 GPUs	96 GB GDDR5		

Build a VM

Create a virtual machine instance

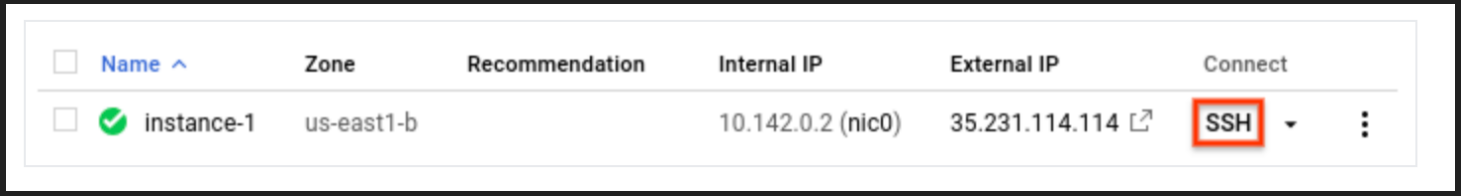
1. In the Cloud Console, go to the **VM instances** page.

   Go to VM instances

2. Click **Create instance**.
3. In the **Boot disk** section, click **Change** to begin configuring your boot disk.
4. On the **Public images** tab, choose **Ubuntu 20.04 LTS**.
5. Click **Select**.
6. In the **Firewall** section, select **Allow HTTP traffic**.
7. Click **Create** to create the instance.

Build a VM

- Once your VM is running, it should show up in your VM Instances dashboard
- You can connect directly to it via SSH by clicking the icon
- Once you're done with the VM, **delete it**



VM Images

- There is an entire library of public pre-built Compute Engine images available!
  - Look into these before you try to install Tensorflow / PyTorch from scratch!
- Involves simply attaching an existing "image" in the form of a storage disk to a brand-new VM

DataProc

- This is where **clusters** are spun up and down
  - Also where jobs are executed on those clusters
- "Hadoop-based clusters" – this is where Spark clusters are born!

Command line utilities

gcloud

- Interacting directly with GCP products
- Spin up VM or cluster, manage instances
- Delete clusters
- Can do everything the web UI can **and then some**, but definitely harder to use

gsutil

- Low-level file management, permissions, access control
- Useful for moving files around

pip-installable with easy environment integration and autocomplete

Need command line utility spin up dask clusters on DataProc

```
gcloud dataproc clusters create ${CLUSTER_NAME} \
  --region ${REGION} \
  --zone ${ZONE} \
  --master-machine-type n1-standard-16 \
  --worker-machine-type n1-standard-16 \
  --image-version preview-ubuntu \
  --optional-components JUPYTER \
  --initialization-actions gs://goog-dataproc-initialization-actions-${REGION}/dask/dask.sh \
  --metadata dask-runtime=yarn \
  --enable-component-gateway
```

https://cloud.google.com/blog/products/data-analytics/improve-data-science-experience-using-scalable-python-data-processing

Amazon Web Services

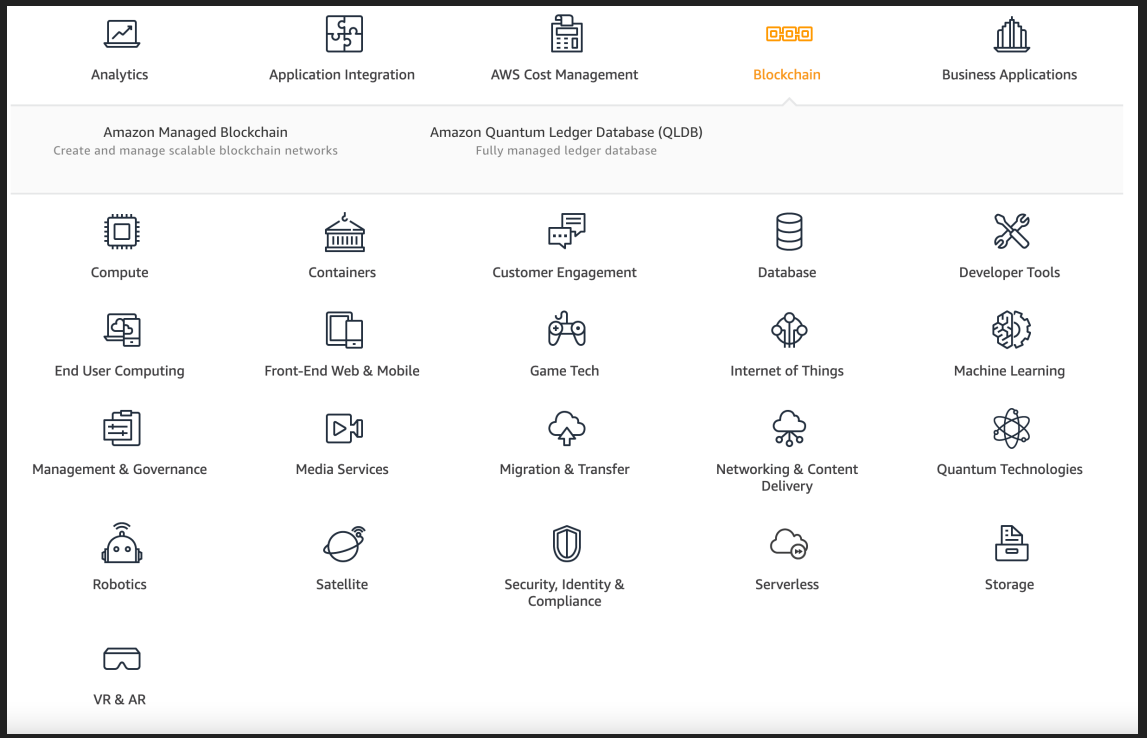
https://aws.amazon.com/

Amazon Web Services

- Home to CSCI 8360 in fall 2016
- Moved to GCP after that, following some high-profile AWS key thefts from GitHub project repos O_O
  - **DON'T PUT YOUR AUTH KEYS IN CODE THAT IS COMMITTED TO PUBLIC REPOS**
- Also moved to GCP because AWS places considerably less emphasis on education vs research ¯_(ツ)_/¯

AWS Product Lineup

- Comparable to GCP



AWS and GCP

- Plenty of equivalencies between the two

- GCP Compute Engine == AWS EC2 (Elastic Compute Cloud)
- GCP DataProc == AWS EMR (Elastic MapReduce)
- GCP Storage == AWS S3 (Simple Storage Solution)
- GCP gcloud + gsutil == AWS awscli (also pip installable!)

CSCI 8360 GCP Storage

- All datasets for DSP spring 2021 projects will be stored here:

gs://uga-dsp/