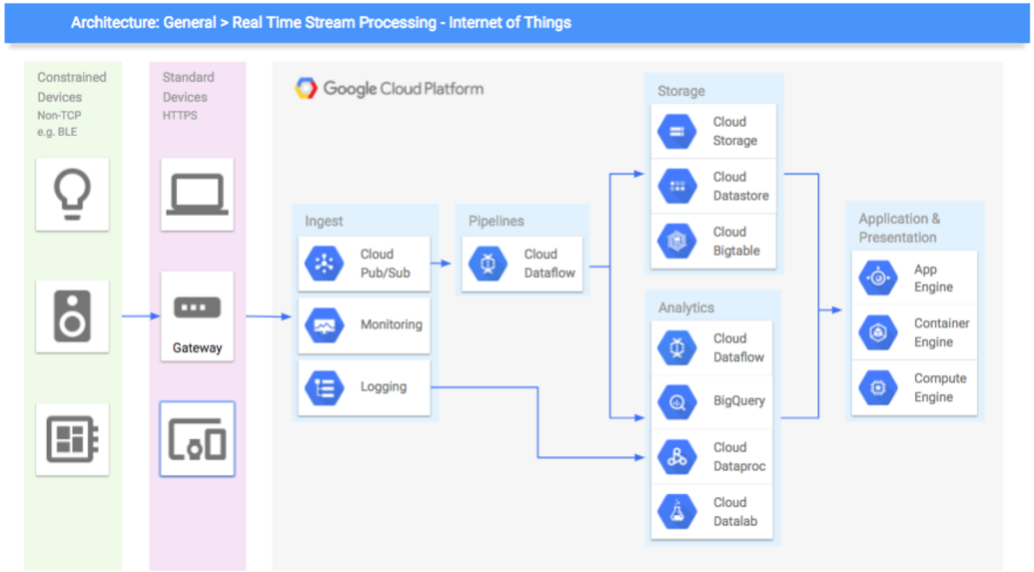
COLUMBIA UNIVERSITY
Information Technology

Intro to Cloud Computing for Research

A Foundations for Research Computing Workshop
By CUIT Research Computing Services

What is the Cloud?

The cloud is a collective pool of shared, on-demand, highly configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned, released, and changed with less management than physical equipment.



Elements of Cloud Computing

On Demand Self-Service

- You can log in and interact with cloud services without engaging with IT.

Broad Network Access

- You can generally access cloud services from anywhere with a network connection.

Resource Pooling

- Storage, memory, CPU processes, and network bandwidth are pooled and assigned to multiple consumers to reduce costs.

Elastic Resources

- Cloud cost saving also relies on elastic resource usage, relinquishing resources when not used and adding resources when in demand.

Pay as you go Model

- All cloud services are measured and metered, which add to the costs.

Benefits of Cloud Computing

Potential Cost Saving

- Certain types of workloads can sometimes be cheaper on the cloud
- Interruptible and Elastic workloads can reduce costs

Reliability

- Due to the sheer amount of resources available cloud can avoid extensive downtimes by quickly re-deploying infrastructure, which - in a physical data center - may require time to troubleshoot and repair.

Portability & Mobility

- Advancements in containerization and cloud management tools make it faster to deploy services.

Types of Cloud Services

Infrastructure-as-a-Service (IaaS)

- IaaS is our primary focus in this workshop. IaaS can be thought of as the big cloud providers such as Amazon Web Services (AWS) Microsoft Azure, and the Google Cloud Platform (GCP).

Platform-as-a-Service (PaaS)

- PaaS includes developer and operations tools, such as the AWS Elastic Beanstalk, Google App Engine, Cloud SQL system, etc. These are basically managed services within an infrastructure environment to make the administration and application of services more efficient.

Software-as-a-Service (SaaS)

- These are cloud applications and tools where the infrastructure is obfuscated from the users, like Google Drive.

Public & Private Cloud

We are focusing on **Public Cloud** options:

Amazon Web Services (AWS)

Google Cloud Platform (GCP)

Microsoft Azure

The above services are providers of cloud infrastructure that's available for public consumption. **Private Cloud** would be an on-premise cloud system, which Columbia does not currently offer.

Cloud at Columbia

Amazon Web Services

Users can sign up for a Columbia AWS account to benefit from features like Columbia's Data Egress waiver, the Columbia & AWS BAA, and directly using a chartstring. The sign up can be found here:

https://cuit.columbia.edu/aws

Cloud at Columbia

Microsoft Azure

Columbia has an agreement with Microsoft Azure but for CUIT it is used mostly for internal infrastructure - disaster recovery and Windows Desktop Virtual Machines. While the Windows Desktop VMs can be made available to researchers Azure is generally not recommended for research at this time.

Cloud at Columbia

Google Cloud Platform

In 2020 Columbia signed a BAA with Google and launched the GCP portal in October 2020, allowing researchers to set up projects under the Columbia domain, log in with their Columbia Gmail accounts, and benefit from general monitoring. The sign up can be found here:

https://cuit.columbia.edu/gcp

Google Cloud Platform

The Google Cloud Platform is the recommended platform for research at Columbia. This tutorial will cover creating an account, navigating the interface, and some potentially useful walkthrough examples of using GCP.

Setting up an account:

Please submit our request form to create your GCP project.

https://cuit.columbia.edu/gcp

The current practice is to request a project from CUIT, who will set up a new project for users. This may create some differences in the general practice of users who may want to create multiple projects for testing purposes.

Accessing and Managing Resources

There are two primary ways to interact with GCP resources - via the command line (gcloud) or the GCP Console (which includes tools to open a command line interface). The GCP Console also contains a cloud shell to use a command line interface from within the console itself.

Accessing and Managing Resources

Command Line

- To manage GCP resources via a terminal or command line interface you must install the **Google Cloud SDK** and **gcloud CLI** tool in your terminal.
- The command line interface allows for powerful, fast access and the ability to launch and manage new resources via the gcloud utility and kubernetes tools.
- The Cloud SDK is useful but for ease of explanation this document will focus on the GCP Console. For quickstarts and setup information please refer to Google's documentation: https://cloud.google.com/sdk/docs/quickstart

Accessing and Managing Resources

Console

- The way to interact with GCP via a graphical interface (GUI) is through the **GCP Console**, at https://console.cloud.google.com/.
- The console provides a clean, organized layout with customizable display as well as billing and monitoring feedback, statistics, and management tools.
- The console also provides explanatory information and even tips and tutorials for getting started with Google products.

GCP Projects

Google organizes workloads and resources under “Projects.”

Projects are the “basis for creating, enabling, and using all Google Cloud services including managing APIs, enabling billing, adding and removing collaborators, and managing permissions for Google Cloud resources.”

A Project consists of three identifiers:

1. An editable Project Name
2. An un-editable Project ID
3. A Unique Project Number generated by Google

Projects at Columbia are created by CUIT for research projects.

Creating a Project

NOTE: This is for vanilla GCP but most will be controlled by CUIT.

1. Log into the GCP Console (https://console.cloud.google.com/home)
2. Go to the Manage resources page in the Cloud Console
3. On the Select organization drop-down list at the top of the page, select the organization in which you want to create a project. If you are a free trial user, skip this step, as this list does not appear.
4. Click Create Project.
5. In the New Project window that appears, enter a project name and select a billing account as applicable.
    a. A project name can contain only letters, numbers, single quotes, hyphens, spaces, or exclamation points, and must be between 4 and 30 characters.
    b. For example, a project called “learning-gcp”
6. Enter the parent organization (Columbia.edu) or folder in the Location box. That resource will be the hierarchical parent of the new project.
7. When you're finished entering new project details, click Create.

Deleting a Project

NOTE: This may be only possible with CUIT assistance.

The best way to avoid charges when you've completed a project is to **delete it**, as this will delete all associated resources with the project.

To shut down a project using the Cloud Console:

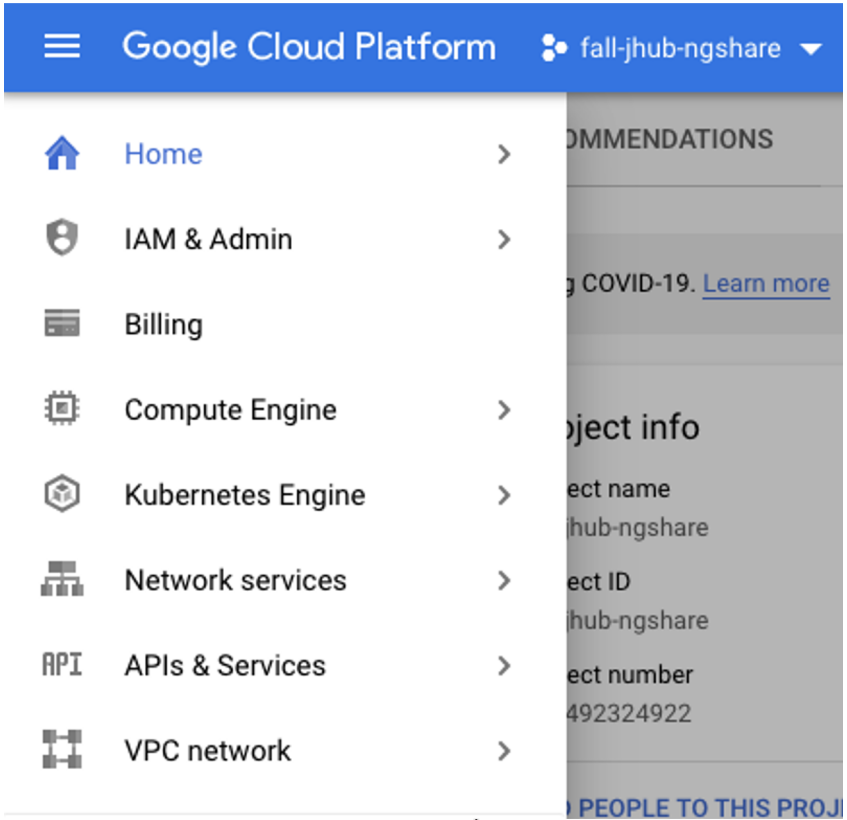
1. Open the **Settings** page (found under **IAM & admin**) in the Google Cloud Console.
2. Click **Select** a project.
3. Select a project you want to delete, and click **Open**.
4. Click **Shut down**.
5. Enter the **Project ID**, then click **Shut down**.

A Brief Introduction to the GCP Console

When you log in to GCP you'll be put into the "Home" tab for your project (seen in the top right, in my example you can see "fall-jhub-ngshare" is the name of my selected project). The home menu gives quick insights into usage, such as "%CPU Utilization" over the past hour or day.

The "Hamburger" menu (the three lines in the top left) gives quick access to services and resources, as does the search bar at the top of the page.

You can use the Hamburger menu to get to useful products, particularly Compute, Billing, Network, Kubernetes, and others that will be covered in this tutorial.



COLUMBIA UNIVERSITY
Information Technology

Break

Noteworthy GCP Services

Compute Engine

The Compute Engine is the backbone of basic computing on GCP. It is the center of tools to spring up virtual servers and services. At the most basic point, you can use the Compute Engine to set up a Virtual Machine.

Storage

GCP offers several storage options, but arguably the most used and most simplistic form is a “bucket.” A bucket is a basic container that holds data in Cloud Storage. Another option would be a database.

Big Query

You can use the Google Cloud Console as a visual interface to complete tasks like running queries, loading data, and exporting data. This quickstart shows you how to query tables in a public dataset and how to load sample data into BigQuery using the Cloud Console.

Kubernetes Engine

The Kubernetes Engine - as the name implies - is how to deploy Kubernetes on GCP. Kubernetes is a container-orchestration tool, namely to rapidly set up and coordinate Docker containers. This can be used for multiple applications and is particularly useful for creating scaling resources, such as an autoscaling Jupyterhub service.

(Plus many more)

Compute Engine Demo - Creating a Linux VM

1. In the Cloud Console, go to the **VM Instances** page.
2. Click **Create instance**.
3. For the **Region** (the general regional area of the compute resources) select us-central1 (Iowa).
    a. Region is important to guarantee your data stays in the US and also affects what resources are available to you, as well as the cost, potentially.
4. For the **Zone** select us-central1-a. Zone is more specific than Region, and useful for keeping resources together.
    a. Different zones have different resources available, so if you need specific resources try different Regions and Zones (within the US).
5. For the **Machine configuration** select the “General-purpose” machine family, the E2 series, and the e2-micro machine type.
6. In the **Boot disk** section, click **Change** to begin configuring your boot disk.
7. On the **Public images** tab, choose **CentOS version CentOS 7**.
    a. CentOS is the open source version of the popular RedHat enterprise version, and a good basis for infrastructure - although Ubuntu and Debian are also popular.
8. Click **Select**.
9. In the **Firewall** section, select **Allow HTTP traffic**.
    a. This allows some networking capabilities, much like a normal VM. This may not always be appropriate, but if you need to do things such as download files from the internet you will need to configure some networking.
10. Click **Create** to create the instance.

Compute Engine Demo - Connecting to a VM

You can connect to the VM by navigating to the VM Instances page within the Compute Engine and selecting the “SSH" option under Connect. There are several options, but the easiest is typically to just select “Open in Browser Window” which will open a terminal in your browser.

Within the VM it will act like any other machine. You have root access by default and can install packages with commands such as $ sudo yum install python3.

You will be logged in as your UNI name to whatever you named in the instance (so for me, mw2931@instance-1).

Compute Engine Demo - Shutting Down a VM

To shutdown the instance (after it's created) go to the VM Instances page within the Compute Engine, select the instance, and select "Delete" on the top bar of the VM Instances page.

Storage - Buckets

Buckets are the simplest form of storage available.

We will go through a demonstration of how to create a bucket and upload this picture to it:

Storage Demo - Creating a Bucket

To create a bucket follow these steps:

1. Open the Cloud Storage browser in the Google Cloud Console.
2. Click **Create bucket** to open the bucket creation form.
3. Enter your bucket information and click **Continue** to complete each step:
    a. Enter a unique **Name** for your bucket.
        i. Do not include sensitive information in the bucket name, because the bucket namespace is global and publicly visible.
        ii. See bucket naming requirements.
    b. Choose **Region us-central1 (Iowa)** for **Location type** and **us-central1-a** for **Location**.
    c. Choose **Standard** for **default storage class**.
    d. Choose **Uniform** for **Access control**.
4. Click **Create**.

Storage Demo - Uploading & Downloading Files

Upload an Object

To upload the image above into your new bucket:

1. Right-click on the image above and download it to your computer.
2. In the Cloud Storage browser page, click the name of the bucket that you created.
3. In the **Objects** tab, click **Upload files**.
4. In the file dialog, go to the file that you downloaded and select it.

After the upload completes, you should see the file name and information about the file, such as its size and type.

Download an object

To download the image from your bucket, click Download > file_download.

Storage Demo - Controlling Bucket Access

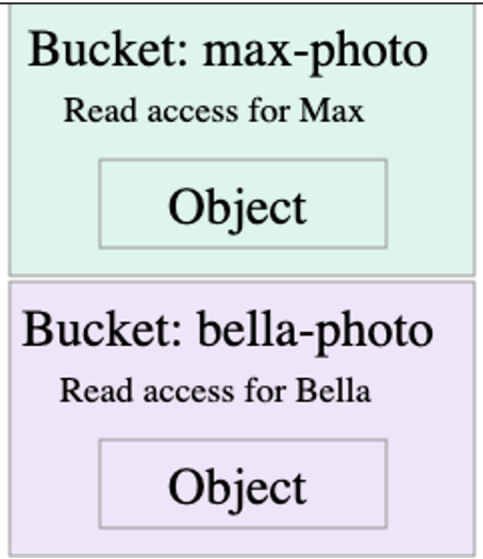
You can set access to buckets in two ways:

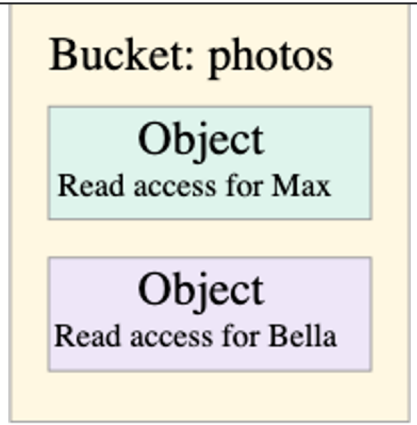
1. **Uniform** access, which is recommended, where access to the entire bucket is guided by the Identity and Access Management (IAM) permissions.
2. **Fine-grained** access, which enables you to use IAM as well as Access Control Lists (ACLs) to manage permissions. The next slide includes a visible outline of the types of access.

You can also **encrypt** data if required.

Storage Demo - Bucket Access Continued

Recommended	Not recommended
Access control: uniform	Access control: fine-grained





Storage Demo - Deleting Buckets

1. Open the Cloud Storage browser in the Google Cloud Console.
2. Select the checkbox of the bucket you want to delete.
3. Click Delete.
4. In the overlay window that appears, confirm you want to delete the bucket and its contents by clicking Delete.

Other Storage Options

Accessing a Bucket from a Compute Instance

There are two ways to access a bucket from a Compute Instance:

1. Use gsutil to interact with the bucket
2. Use gcsfuse to mount the bucket like a volume

These take some setup but are covered in the gsutil Quickstart:

https://cloud.google.com/storage/docs/quickstart-gsutil

Databases

You can also use the Google Cloud Databases for a managed database service.

BigQuery Demo - Query a Public Dataset

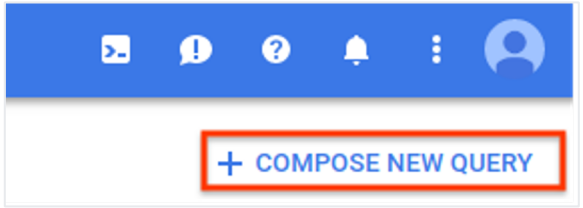
The Cloud Console provides an interface to query tables, including public datasets offered by BigQuery.

In this example, you query the USA Name Data public dataset to determine the most common names in the US between 1910 and 2013.

BigQuery public datasets are displayed by default in the Cloud Console. To open the public datasets project manually, enter the following URL in your browser.

https://console.cloud.google.com/bigquery?p=bigquery-public-data&page=project

BigQuery Demo - Query a Dataset (continued)

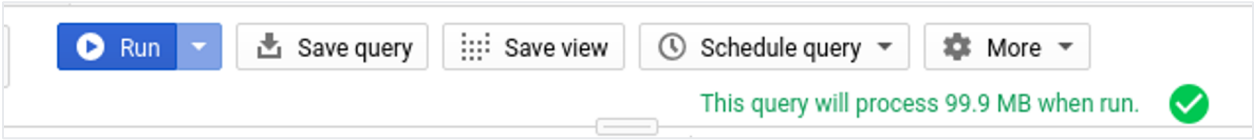


1. Go to the BigQuery page in the Cloud Console
2. Click **Compose new query**. If this text is dimmed, then the **Query editor** is already open.
3. Copy and paste the following query into the query text area.

```
SELECT
  name, gender,
  SUM(number) AS total
FROM
  `bigquery-public-data.usa_names.usa_1910_2013`
GROUP BY
  name, gender
ORDER BY
  total DESC
LIMIT
  10
```

BigQuery Demo - Query a Dataset (continued)

4. To view the query validator, click the green check mark.



If the query is valid, a green check mark appears. If the query is invalid, a red exclamation point appears. If the query is valid, the validator also shows the amount of data the query will process when you run it. The data processed is helpful for determining the cost of running the query.

5. Click **Run**. The query results page appears below the query window. At the top of the query results page, the time elapsed and the data processed by the query are displayed. Below the Query complete... message, a table displays the query results with a header row containing the name of each column you selected in the query.

BigQuery Demo - Query a Dataset (Results)

Query results SAVE RESULTS EXPLORE IN DATA STUDIO

Query complete (1.1 sec elapsed, 99.9 MB processed)

Job information | Results | JSON | Execution details

Row	name	gender	total
1	James	M	4924235
2	John	M	4818746
3	Robert	M	4703680
4	Michael	M	4280040
5	William	M	3811998
6	Mary	F	3728041
7	David	M	3541625
8	Richard	M	2526927
9	Joseph	M	2467298
10	Charles	M	2237170

BigQuery Demo - Create a Dataset

Download the data

The file you're downloading contains approximately 7 MB of data about popular baby names, and it is provided by the US Social Security Administration.

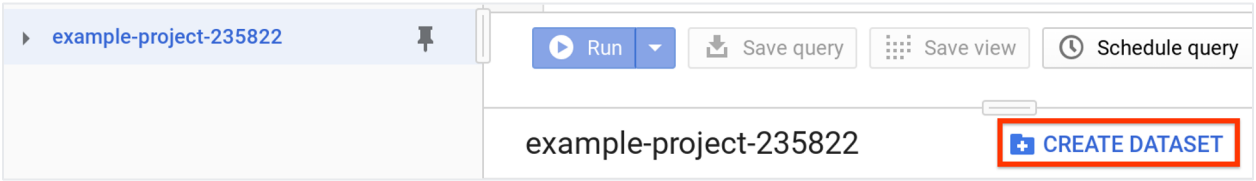
1. Download the baby names zip file.
2. Extract the file onto your machine.
   The zip file contains a `NationalReadMe.pdf` file that describes the dataset.
   Learn more about the dataset.
3. Open the file named `yob2014.txt` to see what the data looks like. The file is a comma-separated value (CSV) file with the following three columns: name, sex (`M` or `F`), and number of children with that name. The file has no header row.
4. Note the location of the `yob2014.txt` file so that you can find it later.

BigQuery Demo - Create a Dataset (continued)

Create a dataset

Next, create a dataset in the Cloud Console to store the data.

1. If necessary, open the BigQuery page in the Cloud Console.
2. In the navigation panel, in the **Resources** section, click your project name.
3. On the right side, in the details panel, click **Create dataset**.

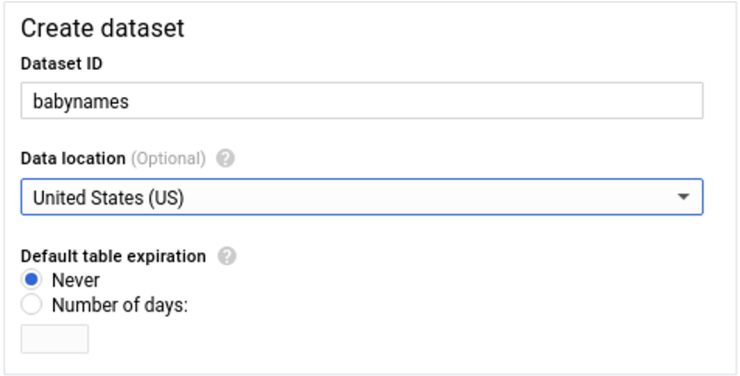


BigQuery Demo - Create a Dataset (continued)

4. On the **Create dataset** page, do the following:

   - For **Dataset ID**, enter babynames.
   - For **Data location**, choose **United States (US)**. Currently, the public datasets are stored in the US multi-region location. For simplicity, place your dataset in the same location.

5. Leave all of the other default settings in place and click Create dataset.

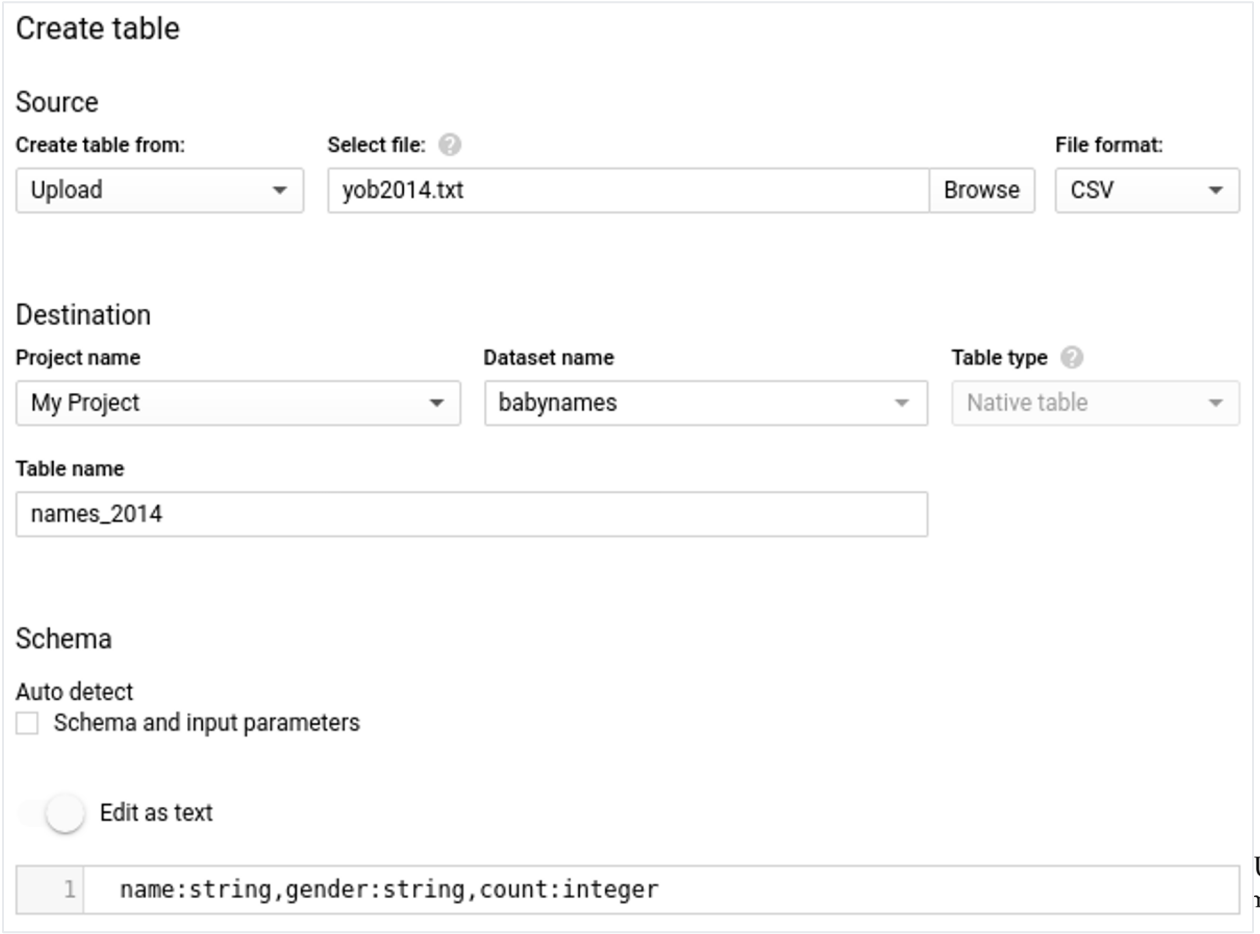


BigQuery Demo - Create a Dataset (continued)

Load the data into a new table

1. In the navigation panel, in the **Resources** section, click the **babynames** dataset that you just created.
2. On the right side, in the details panel, click **Create table**.
   Use the default values for all settings unless otherwise indicated.
3. On the **Create table** page, do the following:
   - For **Source**, click **Empty table** and choose **Upload**.
   - For **Select file**, click **Browse**, navigate to the yob2014.txt file, and click **Open**.
   - For **File format**, click **Avro** and choose **CSV**.
   - In the **Destination** section, for **Table name**, enter names_2014.
   - In the **Schema** section, click the **Edit as text** toggle and paste the following schema definition into the box.
     name:string,gender:string,count:integer

BigQuery Demo - Create a Dataset (continued)



BigQuery Demo - Create a Dataset (continued)

4. Click **Create Table**

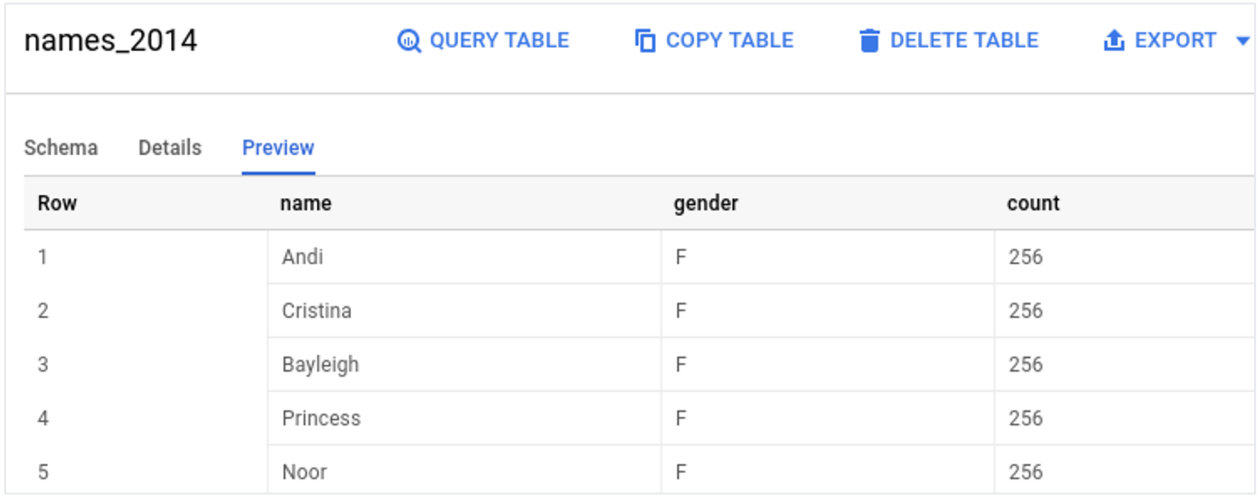
5. Wait for BigQuery to create the table and load the data. While BigQuery loads the data, a (1 running) string displays beside the job history in the navigation panel. The string disappears after the data is loaded.

BigQuery Demo - Create a Dataset (continued)

Preview the table

After the **(1 running)** string disappears, you can access the table. To preview the first few rows of the data, follow these steps:

1. In the navigation panel, select **babynames > names_2014**.
2. In the details panel, click the **Preview** tab.



Row	name	gender	count
1	Andi	F	256
2	Cristina	F	256
3	Bayleigh	F	256
4	Princess	F	256
5	Noor	F	256

BigQuery Demo - Create a Dataset (continued)

Query the table

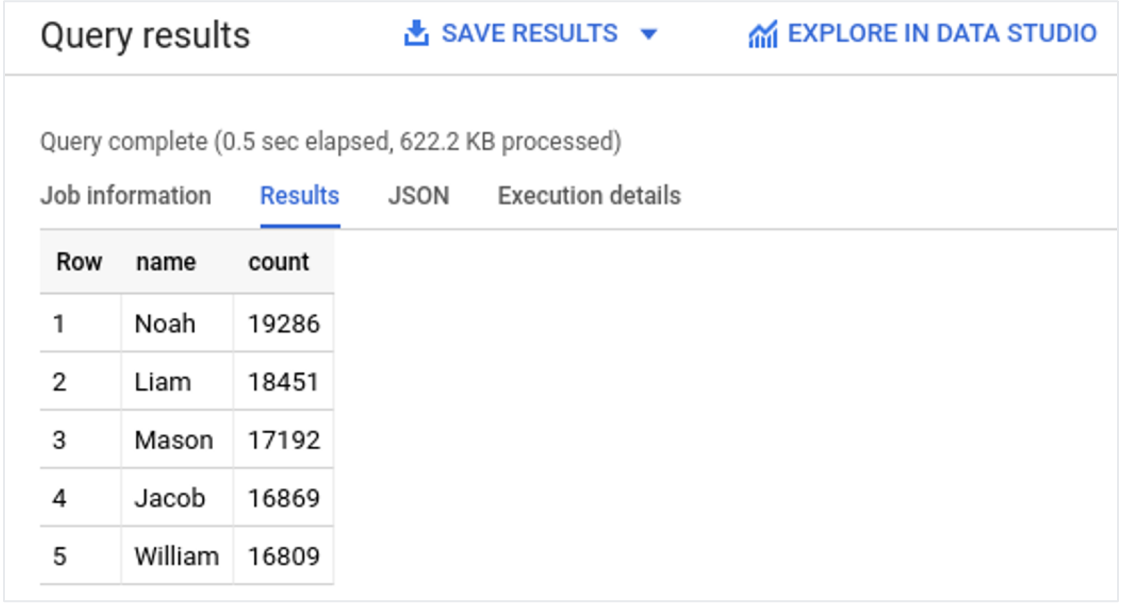
Now that you've loaded data into a table, you can query it. The process is identical to the previous example, except that this time, you're querying your table instead.

1. If necessary, click the **Compose new query** button. Unless you hid the query window previously, it should still be visible.
2. Copy and paste the following query into the query text area. This query retrieves the top five baby names for US males in 2014.

```
SELECT
  name,
  count
FROM
  `babynames.names_2014`
WHERE
  gender = 'M'
ORDER BY
  count DESC
LIMIT
  5
```

BigQuery Demo - Create a Dataset (continued)

Click **Run**. The results are displayed below the query window.



Row	name	count
1	Noah	19286
2	Liam	18451
3	Mason	17192
4	Jacob	16869
5	William	16809

BigQuery Demo - Clean Up

To avoid incurring charges to your Google Cloud account for the resources used in this quickstart, follow these steps.

1. If necessary, open the BigQuery page in the Cloud Console.
2. In the navigation panel, in the **Resources** section, click the **babynames** dataset you created.
3. In the details panel, on the right side, click **Delete dataset**. This action deletes the dataset, the table, and all the data.
4. In the **Delete dataset** dialog box, confirm the delete command by typing the name of your dataset (`babynames`) and then click **Delete**.

Break

Cloud Usage and Cost Monitoring

To view the Cloud Billing reports for your Cloud Billing account:

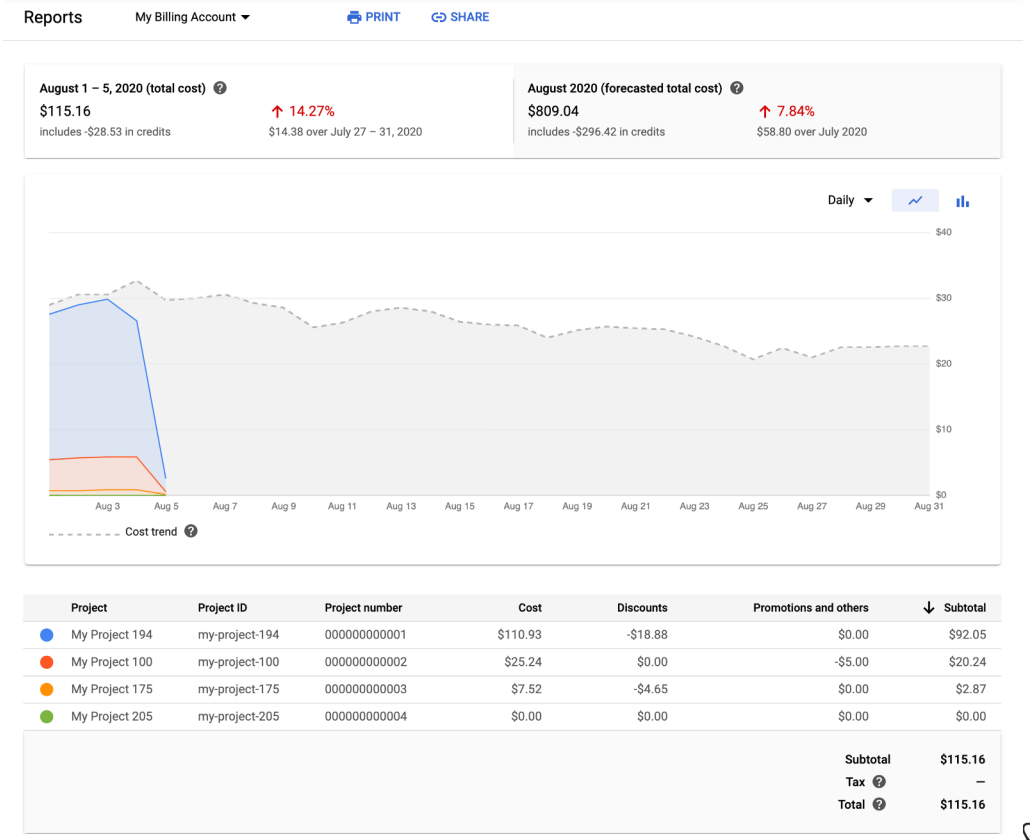
1. Open the console **Navigation menu** (menu), and then select **Billing**.
   If you have more than one Cloud Billing account, do one of the following:
   - To manage Cloud Billing for the current Cloud project, select **Go to linked billing account**.
   - To locate a different Cloud Billing account, select **Manage billing accounts** and choose the account for which you'd like to see reports.
2. In the Billing navigation menu, select **Reports**.

Cloud Usage and Cost Monitoring

The **default view** uses the **Current month, all projects** preset. In the default view:

- The report displays the current month's daily usage-specific costs grouped by project (for all Google Cloud services), inclusive of any usage-specific credits applied. The report shows projects that incurred usage-specific costs for the month.
- The summary bar above the chart provides a split view of cost: actual cost-to-date for the current month, and the total forecasted cost for the entire current month.
- A cost trend line in the chart indicates the direction your forecasted cost is trending.
- Each line in the chart (and row in the summary table) corresponds to the project, ranked largest to smallest by cost.
- The summary footer, below the table, displays the cost breakdown based on the filter selections chosen.

Cloud Usage and Cost Monitoring



COLUMBIA UNIVERSITY
Information Technology

Cloud Usage and Monitoring

Chart Setting Options

Time aggregation - You can specify a different time aggregation using the Daily/Monthly list at the top right of the chart.

Chart Style - You can specify a different chart display style using the Line Chart/Bar Chart selector at the top right of the chart.

Data Order - The order of the data displayed in the stacked line or bar chart is controlled by the data category you are sorting on, as indicated by an arrow in the column header in the summary table. You can change the sort order of the data by clicking on a different column header. The direction of the arrow indicates if you are sorting largest to smallest (down arrow arrow_downward) or smallest to largest (up arrow arrow_upward). To reverse the sort order on the selected column, click the column header again.

Cloud Usage and Monitoring

View Options

- Preset Views
- Time Ranges
- Grouping
- Filters
- Forecasting

Cloud Usage and Monitoring

Cost Saving Possibilities

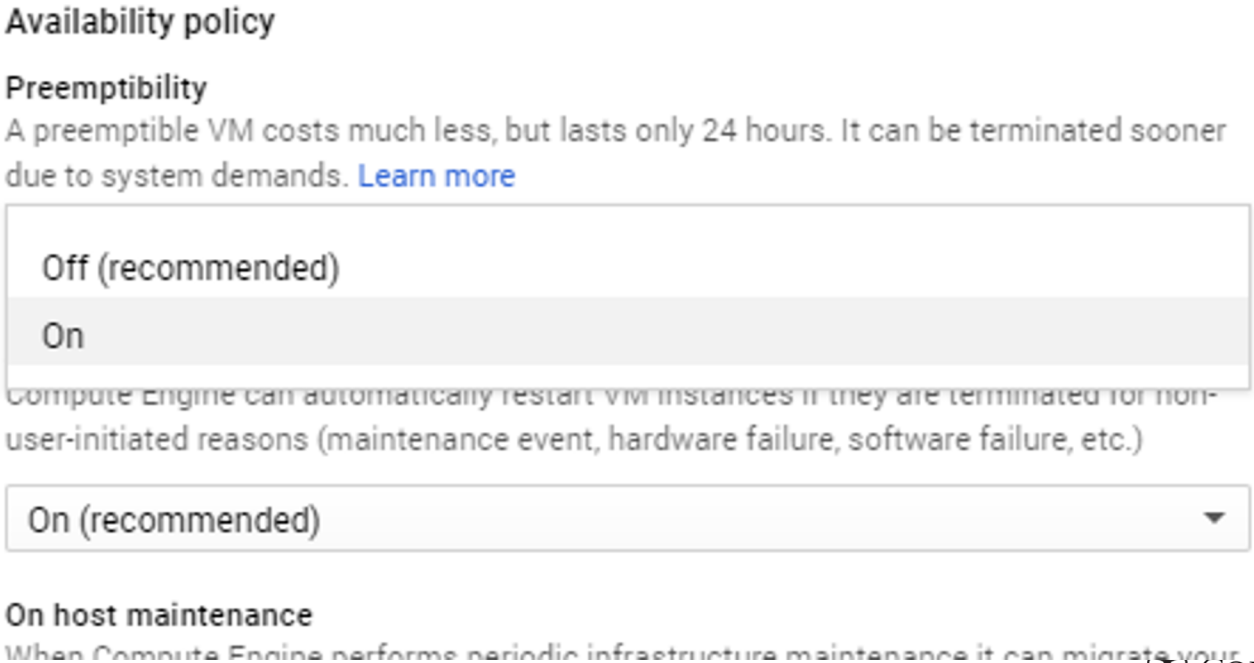
Preemptible Virtual Machines

A preemptible VM is an instance that you can create and run at a lower price than normal instances. However, Compute Engine might stop (preempt) these instances if it requires access to those resources for other tasks. Preemptible instances are excess Compute Engine capacity, so their availability varies with usage.

This means that you are best when you have a workflow that can be disrupted, or choose small machines that are less likely to be preempted - or during non-peak hours.

Creating Preemptible VMs

You can make a VM preemptible by selecting the option in your Instance Template of the VM.



Stopping Charges

The only surefire way to stop all charges for a project is to **delete the project.**

If this is not possible, you can go step by step through the services that are being charged (which can be seen in the Billing report) and delete associated resources.

Instances that are in a **TERMINATED** state are not charged for per-second usage and do not count toward your regional CPU quota, so you can choose to stop instances that you are not using, saving you from being charged for instances that aren't active. After you are ready, you can come back and start the same instances again, with the same instance properties, metadata, and resources.

While instances are not charged for per-second usage while in the TERMINATED state **but any resources attached to the virtual machine, such as static IPs and persistent disks, are charged until they are deleted.**

Stopping Charges - Compute

1. In the Cloud Console, go to the VM instances page.
2. Select one or more instances that you want to stop.
3. Click Stop.

After the instance is in the TERMINATED state, you can restart the instance or delete it. You can also leave an instance in a TERMINATED state indefinitely. However, if you do not plan to restart the instance, delete it instead.

Stopping Charges - Storage

1. Open the Cloud Storage browser in the Google Cloud Console.
2. Select the checkbox of the bucket you want to delete.
3. Click Delete.
4. In the overlay window that appears, confirm you want to delete the bucket and its contents by clicking Delete.

Stopping Charges - Networking

Make sure you release Static IPs:

1. In the Cloud Console, go to the External IP addresses page.
2. Check the box next to the IP address to release.
3. Click Release IP address.

Stopping Charges - Kubernetes

1. Visit the Google Kubernetes Engine menu in Cloud Console.
2. Click the Delete icon (it looks like a trash can) next to the cluster you want to delete.
3. When prompted to confirm, click Delete again.

End of Day 1

COLUMBIA UNIVERSITY
Information Technology

Intro to Cloud Computing for Research

A Foundations for Research Computing Workshop
By CUIT Research Computing Services

Machine Learning Tools & Notebooks (AI Platform)

There are a lot of ways to do computational computing. You could set up your own JupyterLab server or RStudio server, for instance. GCP has a tool for managed JupyterLab notebooks, however - the AI Platform.

The AI Platform allows users to get up and running quickly, scale on demand, and integrate with other tools such as BigQuery. The AI Platform provides a pre-configured environment that supports the most popular data science libraries, including R, pandas, NumPy, SciPy, scikit-learn, and Matplotlib, and ML frameworks like TensorFlow, Keras, fast.ai, RAPIDS, XGBoost, and PyTorch.

AI Platform Demo - Create a New Notebook Instance

Follow the steps in Before you begin to create a Google Cloud (Google Cloud) project and enable the AI Platform Notebooks API. The following creates a notebook with default properties.

1. Go to the **AI Platform Notebooks** page in the Google Cloud Console.
2. Click **New Instance**, select an instance type, and then choose whether to include a GPU.
    a. You can also use **Customize instance** to change things like the version of **R and Python.**
    b. Note: not all the options can be used with GPU, but Tensorflow and Pytorch can.
3. If you choose to include a GPU, you must select the option to **Install NVIDIA GPU driver automatically for me**. You can adjust the number of GPUs later if you need to. For information on adjusting the number of GPUs, see Manage hardware accelerators for a notebook.
    a. If you want to change the default boot disk settings or encryption settings, expand the **Boot disk** section - we will not do this now
4. Click **Create**.
5. AI Platform Notebooks creates a new instance based on your selected framework. An **Open JupyterLab** link becomes active when it's ready to use.
6. Grant access using the IAM settings - editors of a GCP Project can access the notebook.

AI Platform Demo - Access Your Notebook

Open the notebook

Complete these steps to open a notebook instance:

1. On the AI Platform Notebooks page in the Google Cloud Console, click **Open JupyterLab** to open the notebook.
2. AI Platform Notebooks opens your notebook.

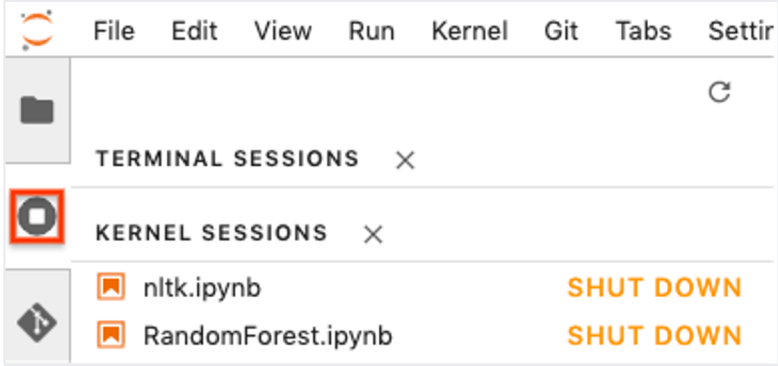
Installing Dependencies

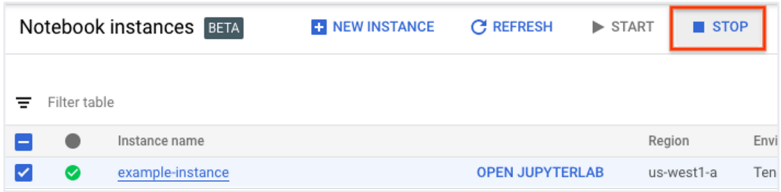
While outside of the scope of this tutorial, you can install dependencies within notebooks. When you open your new notebook, there is a default code cell where you can enter code, in this case Python 3, to install dependencies. For example, to install the mxnet deep learning library in a Python 3 notebook, enter the following in the code cell: `%pip install mxnet`. To install the library in a Python 2 notebook, enter the following: `!pip2 install mxnet`.

AI Platform Demo - Shutting Down a Notebook Instance

To shut down a notebook instance, follow these steps:

1. Go to the **AI Platform Notebooks** page in the Google Cloud Console.
   Go to the AI Platform Notebooks page
2. Select **Open JupyterLab** for the instance that you want to open. It is important to stop all running processes in case there are operations that need to complete before you shut down your notebook instance. For example, I/O processes that are writing to disk.
   To stop all running processes:
   1. Select the **Running Terminals and Kernels** tab to show all of the processes running.
   2. Click **Shutdown** next to each running process.
3. Close the browser tab or window for your notebook instance.
4. Return to the **AI Platform Notebooks** page in the Google Cloud Console.
   Go to the AI Platform Notebooks page





Select the instance that you want to shut down, and then select **Stop**.

GPU Workstations

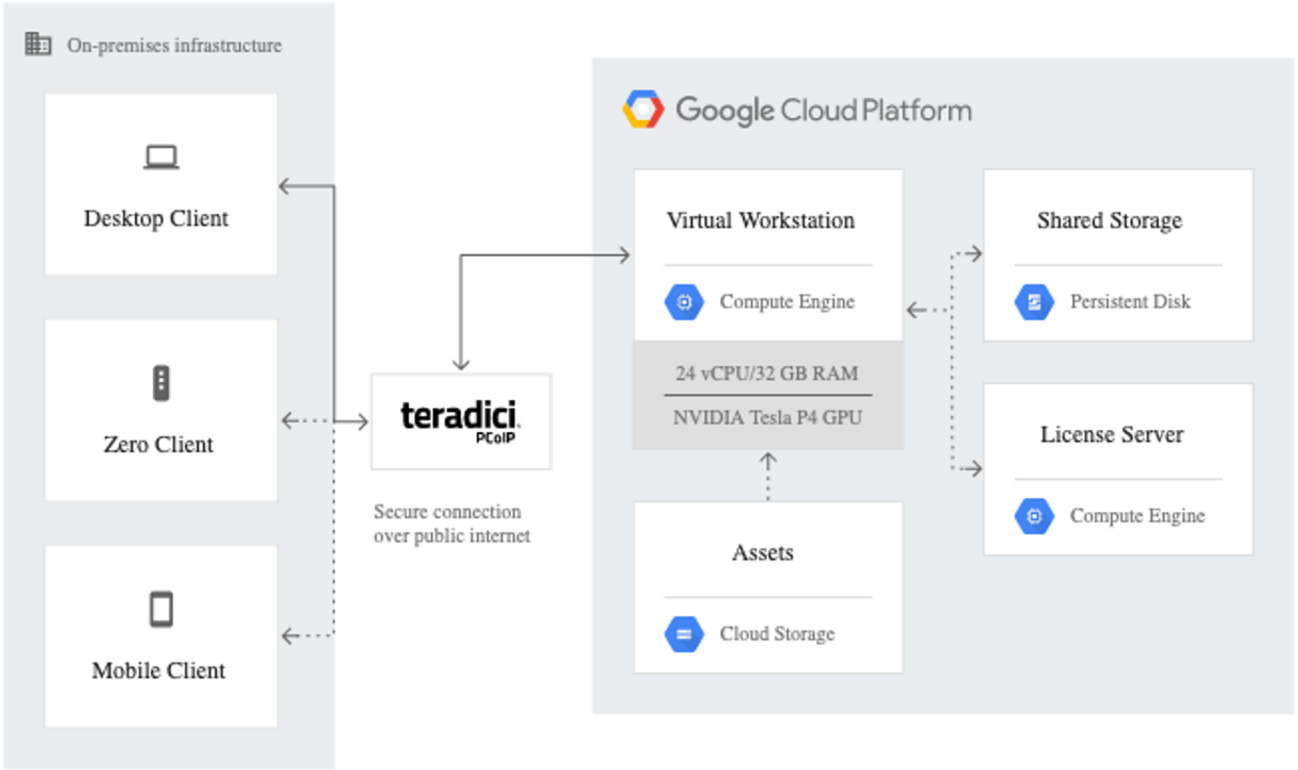
The goal of this example is to create a Compute Engine instance with a GPU that acts as a virtual workstation.

Remember to activate the Compute Engine API if you haven't.

In addition, make sure you have the following:

- A Google Cloud project with quota for virtual workstation GPUs in your selected zone. You can get a list of GPU availability by using the `gcloud compute accelerator-types list` command.
- A Google Chrome browser to access the Cloud Console.
- The Chrome RDP for Google Cloud Chrome extension for initial access to your virtual workstation.
- A Teradici Zero Client or the latest Teradici software client for Windows, Mac, or Linux to access the virtual workstation.
- A Teradici Cloud Access Software license. You can sign up for a trial license, or contact your Teradici representative. You will be provided with a 30-day trial registration code to use for this virtual workstation.

GPU Workstations Demo - Architecture



GPU Workstation Demo - Pick a GPU

1. Open Cloud Shell. (If you're using the Cloud SDK, open a terminal window on your computer.)
   Go to Cloud Shell
2. Get a list of the zones in which GPUs are available:
3. `gcloud compute accelerator-types list`
4. Take note of the zone that's physically closest to you.
5. Set the zone that you want to work with:
6. `gcloud config set compute/zone` **zone**
7. Replace **zone** with the name of the zone you're using, such as `us-central1-b`.

Different GPUs can be attached to different instances. The example in this tutorial consists of a 24-vCPU virtual workstation, which is the maximum number of vCPUs allowed per NVIDIA Tesla P4 GPU.

GPU Workstation Demo - Create the Workstation

Teradici Graphics Agent (which you install on your virtual workstation later in this tutorial) requires you to enable IP forwarding and to allow HTTPS server traffic during virtual workstation creation.

In Cloud Shell, create the Compute Engine virtual workstation instance. You must provide values for the placeholders such as **name**.

GPU Workstation Demo - Create the Workstation

```
gcloud compute instances create name \
   --machine-type machine-type \
   --accelerator
type=accelerator,count=num-gpus \
   --can-ip-forward \
   --maintenance-policy "TERMINATE" \
   --tags "https-server" \
   --image-project windows-cloud \
   --image-family windows-2016 \
   --boot-disk-size size
```

Example:

```
gcloud compute instances create test-vws \
   --machine-type custom-24-32768 \
   --accelerator type=nvidia-tesla-p4-
vws,count=1 \
   --can-ip-forward \
   --maintenance-policy "TERMINATE" \
   --tags "https-server" \
   --image-project centos-cloud \
   --image-family centos-7 \
   --boot-disk-size 100
```

Make a note of the virtual workstation's external IP address in the listing. You will use it later in the tutorial.

GPU Workstation Demo - SSH & Installs

1. In Cloud Shell, connect to the new virtual workstation:

```
gcloud compute ssh test-vws
```

1. Set your account password. Teradici PCoIP requires a user password to be set.

```
sudo passwd `whoami`
```

1. Install Required Components

```
sudo yum -y update
sudo yum -y install kernel-devel
sudo yum -y groupinstall "KDE desktop" "X Window System" "Fonts"
sudo yum -y groupinstall "Development Tools"
sudo yum -y groupinstall "Server with GUI"
```

1. Reboot

```
sudo reboot
```

GPU Workstation Demo - NVIDIA Drivers

NVIDIA T4, NVIDIA Tesla P4, and NVIDIA Tesla P100 GPUs work on Google Cloud only with qualified NVIDIA Quadro Virtual Data Center Workstation (vWS) drivers for both compute and display workloads.

1. Open the **Google Cloud Shell**
2. Reconnect to the workstation
   `gcloud compute ssh test-vws`
1. Get a listing of the latest drivers:
   `gsutil ls gs://nvidia-drivers-us-public/GRID`
1. Install the Driver
   ```
   curl -O \
   https://storage.googleapis.com/nvidia-drivers-us-public/GRID/GRID11.1/NVIDIA-Linux-x86_64-450.80.02-grid.run
   sudo bash NVIDIA-Linux-x86_64-450.80.02-grid.run
   ```

For this tutorial, you use the latest graphics drivers that are available at the time of writing: **GRID11.1 Linux driver (version 450.80.02)**.

GPU Workstation Demo - Install Teradici

1. Verify the driver is installed and working:
   ```
   nvidia-smi
   ```

1. Add the Taradici Software
   ```
   sudo rpm --import https://downloads.teradici.com/rhel/teradici.pub.gpg
   sudo yum -y install wget
   sudo wget -O /etc/yum.repos.d/pcoip.repo \
     https://downloads.teradici.com/rhel/pcoip.repo
   ```

1. Update the repository
   ```
   sudo yum -y update
   ```

1. Install the Cloud Access Software
   ```
   sudo yum -y install pcoip-agent-graphics
   ```

1. Set the display state to graphical
   ```
   sudo systemctl set-default graphical.target
   ```

1. Reboot
   ```
   sudo reboot
   ```

GPU Workstation Demo - Register Teradici

To use Teradici Graphics Agent, you must have a license, as noted earlier in the tutorial.

1. In Cloud Shell, reconnect to the virtual workstation:

   ```
   gcloud compute ssh test-vws
   ```

1. On your virtual workstation, activate your Teradici Cloud Access Software license:

   ```
   pcoip-register-host --registration-code=registration-code
   ```

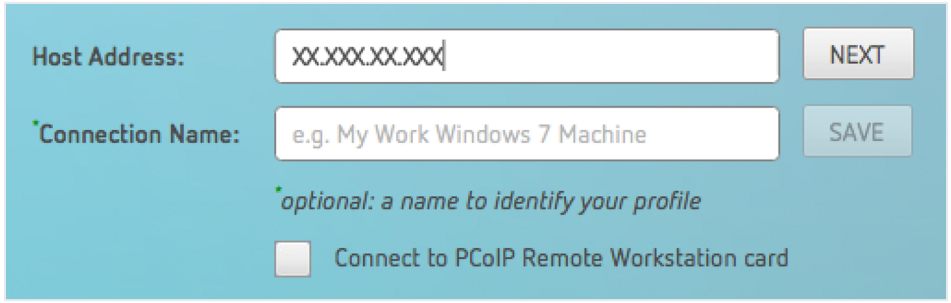
GPU Workstation Demo - Firewall Rules

The PCoIP client communicates with your virtual workstation using several ports. You must set firewall rules that allow traffic to and from your virtual workstation.

- In Cloud Shell on your local computer (not on the virtual workstation), create a firewall rule that opens the required ports:
- gcloud compute firewall-rules create allow-teradici \
- --allow tcp:443,tcp:4172,udp:4172,tcp:60443

GPU Workstation Demo - Sign In

1. On your local computer, go to the PCoIP Clients section on the Teradici support page, and then download, install, and launch the PCoIP Client application for your operating system.
2. Select **New Connection**.
3. In the **Host Address** field, enter the external IP address of your virtual workstation. If you want, you can enter a name for the connection.
4. When you are connected, authenticate by entering the username and password that you created earlier for the virtual workstation.
5. Select the desktop to run and then click **Connect**.
    a. In a few seconds, you see your Linux desktop.



GPU Workstation Demo - Deleting the Instance

This is the same as shutting down and deleting any Compute instance.

1. In the Cloud Console, go to the VM instances page.
2. Select one or more instances that you want to stop.
3. Click Stop.

After the instance is in the TERMINATED state, you can restart the instance or delete it. You can also leave an instance in a TERMINATED state indefinitely. However, if you do not plan to restart the instance, delete it instead.

Kubernetes Engine Demo

Take the following steps to enable the Kubernetes Engine API:

1. Visit the Kubernetes Engine page in the Google Cloud Console.
2. Create or select a project.
3. Wait for the API and related services to be enabled. This can take several minutes.
4. Make sure that billing is enabled for your Google Cloud project. Learn how to confirm billing is enabled for your project.
    - You may need to confirm your quota in the Cloud Console to ensure you have 1 Compute Engine CPU permitted in your cluster's region and 1 In-Use IP address.

Kubernetes Engine Demo - Enter the Cloud Shell

To launch Cloud Shell, perform the following steps:

1. Go to Google Cloud Console.
2. From the upper-right corner of the console, click the Activate Cloud Shell button:

A Cloud Shell session opens inside a frame lower on the console. You use this shell to run gcloud and kubectl commands.

Kubernetes Engine Demo - Set Default Zones

Setting Projects and Zones

Run the following command, replacing **project-id** with your project ID:

```
gcloud config set project project-id
```

Setting a default compute zone

Run the following command, replacing **compute-zone** with your compute zone, such as us-central1-a:

```
gcloud config set compute/zone compute-zone
```

Kubernetes Engine Demo - Creating a GKE Cluster

A *cluster* consists of at least one *cluster control plane* machine and multiple worker machines called *nodes*. Nodes are Compute Engine virtual machine (VM) instances that run the Kubernetes processes necessary to make them part of the cluster. You deploy applications to clusters, and the applications run on the nodes.

The following command creates a one-node cluster. Replace **cluster-name** with the name of your cluster:

```
gcloud container clusters create cluster-name --num-nodes=1
```

Kubernetes Engine Demo - Get Authentication

After creating your cluster, you need to get authentication credentials to interact with the cluster:

```
gcloud container clusters get-credentials cluster-name
```

This command configures kubectl to use the cluster you created.

Kubernetes Engine Demo - Deploying a Containerized Application to the Cluster

1. Creating the Deployment
    a. To run hello-app in your cluster, run the following command:

```
kubectl create deployment hello-server --image=gcr.io/google-samples/hello-app:1.0
```

1. Exposing the Deployment
    a. You need to expose it to the internet so that users can access it.

```
kubectl expose deployment hello-server --type LoadBalancer \
 --port 80 --target-port 8080
```

Kubernetes Engine Demo - Inspecting and viewing the application

1. Inspect the running Pods by using kubectl get pods:
2. kubectl get pods
3. You should see one hello-server Pod running on your cluster.
4. Inspect the hello-server Service by using kubectl get service:
5. kubectl get service hello-server
6. From this command's output, copy the Service's external IP address from the EXTERNAL-IP column.
   Note: You might need to wait several minutes before the Service's external IP address populates. If the application's external IP is **<pending>**, run **kubectl get** again.
7. View the application from your web browser by using the external IP address with the exposed port:
8. http://**external-ip**/

You have just deployed a containerized web application to GKE.

Kubernetes Engine Demo - Clean Up

To avoid incurring charges to your Google Cloud account for the resources used in this quickstart, follow these steps.

1. Delete the application's Service by running kubectl delete:
2. kubectl delete service hello-server
3. This command deletes the Compute Engine load balancer that you created when you exposed the Deployment.
4. Delete your cluster by running gcloud container clusters delete:
5. gcloud container clusters delete **cluster-name**

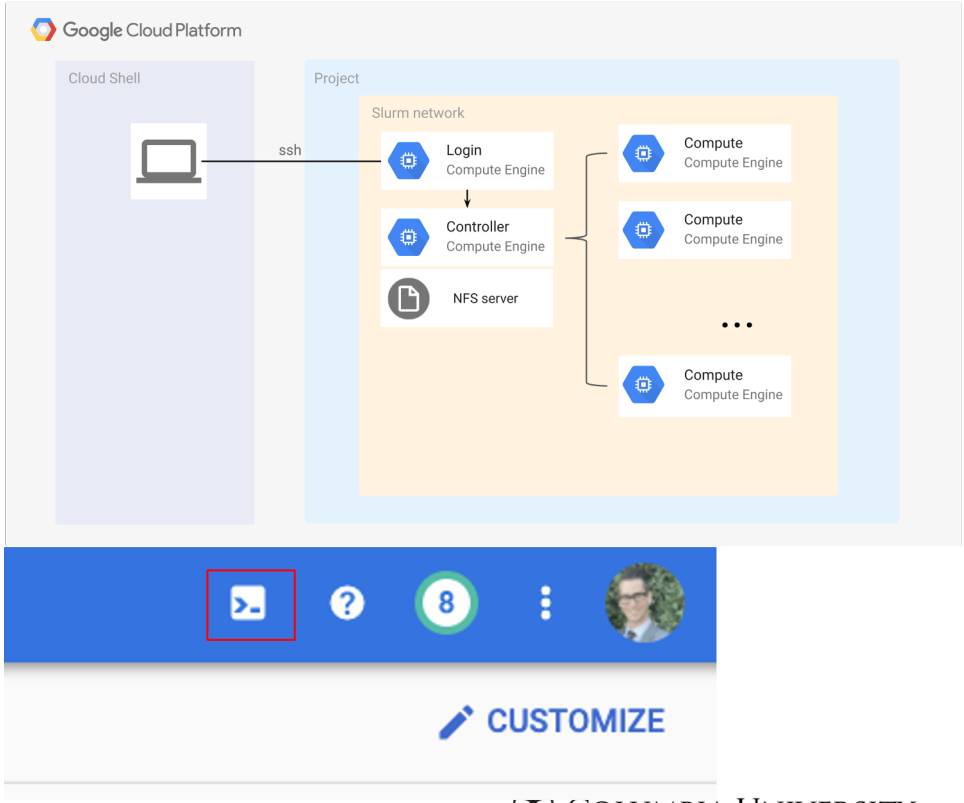
Slurm Cluster Demo - The Basics

Source

This demo will cover the basics of deploying a simple autoscaling Slurm cluster on Compute Engine.

Before you begin, select a project and navigate to the Compute Engine section to enable the Compute Engine API.

The work will be conducted via the **Cloud Shell**, so activate it once you're ready (as seen to the right)



Slurm Cluster Demo - Setup

In the Cloud Shell create a new folder called slurm-cluster and enter it

```
$ mkdir slurm-cluster
$ cd slurm-cluster
```

Then create an environment variables folder, such as source.txt and define the following

```
export CLUSTER_DEPLOY_NAME="cluster-deployment-name"
export CLUSTER_NAME="cluster-name"
export CLUSTER_REGION="cluster-region"
export CLUSTER_ZONE="cluster-zone"
```

Then activate these with `$ source source.txt`

Slurm Cluster Demo - Setup

Clone the slurm-gcp Github repository:

```
$ git clone https://github.com/SchedMD/slurm-gcp.git
```

In the slurm-gcp folder, copy the slurm-cluster.yaml file to the ${CLUSTER_DEPLOY_NAME}.yaml file:

```
cd slurm-gcp
cp slurm-cluster.yaml ${CLUSTER_DEPLOY_NAME}.yaml
```

Slurm Cluster Demo - Modify Deployment YAML

In a text editor, modify and save the ${CLUSTER_DEPLOY_NAME}.yaml file for your environment. Use the types as defined in the slurm.jinja.schema file, which specifies default and permissible values for all configuration properties except the default_users value. Make the following required changes:

- cluster_name: change the name of your cluster to be **cluster-name**.
- region and zone: replace with **cluster-region** and **cluster-zone**
- compute_machine_type: (Optional) to use a different machine type, change the compute_machine_type value. For example, if you need more CPU cores or memory than are available with the default choice of n1-standard-2, choose n1-standard-4. For more information, see Machine types.
- vpc_net and vpc_subnet: (Optional) use an existing Virtual Private Cloud (VPC) network and VPC subnet. The network and subnet requirements are described in the slurm.jinja.schema file. If you don't specify values, a new network or subnet is created for your cluster. For more information, see VPC networks.

Slurm Cluster Demo - Modify Deployment YAML

You can customize other aspects of the cluster within the `${CLUSTER_DEPLOY_NAME}.yaml` file, such as how many nodes are always up and available (static_node_count) the autoscaling limits (max_node_count) which limits how many nodes can be spun up dynamically. These additional nodes beyond the static nodes are ephemeral nodes, that are created for jobs and shortly after completing their task are destroyed, presuming that there are no other jobs requesting them.

The ephemeral nodes are a good way to potentially save costs.

You can also set up different Slurm Partitions, including what kind of CPUs are available or if GPUs are available. This can be useful for different workload types.

Slurm Cluster Demo - Deploy Slurm

In Cloud Shell, use Deployment Manager to deploy your cluster to Google Cloud:

```
gcloud deployment-manager deployments \
   --project="$(gcloud config get-value core/project)" \
   create $CLUSTER_DEPLOY_NAME \
   --config ${CLUSTER_DEPLOY_NAME}.yaml
```

The cluster configuration takes 5-10 minutes to complete. Track the progress of the configuration:

```
gcloud compute ssh ${CLUSTER_NAME}-controller \
   --command "sudo journalctl -fu google-startup-scripts.service" \
   --zone $CLUSTER_ZONE
```

To stop watching the configuration, press Control+C.

Slurm Cluster Demo - Verify Cluster

In Cloud Shell, check that the cluster is ready by logging in to the login node:

```
export CLUSTER_LOGIN_NODE=$(gcloud compute instances list \
    --zones ${CLUSTER_ZONE} \
    --filter="name ~ .*login." \
    --format="value(name)" | head -n1)
gcloud compute ssh ${CLUSTER_LOGIN_NODE} \
    --zone $CLUSTER_ZONE
```

It will display a Slurm graphic in ASCII text when complete.

To connect to the cluster you can use the command:

```
gcloud compute ssh ${CLUSTER_LOGIN_NODE}  --zone $CLUSTER_ZONE
```

To exit the cluster, press `Control+D`

Slurm Cluster Demo - Verify Cluster

When the cluster is ready, schedule a job to verify that it is working correctly. This job runs the `hostname` command on several nodes in the cluster.

```
gcloud compute ssh ${CLUSTER_LOGIN_NODE} \
   --command 'sbatch -N2 --wrap="srun hostname"' --zone $CLUSTER_ZONE

gcloud compute ssh ${CLUSTER_LOGIN_NODE}\
   --command 'cat slurm-*.out'  --zone $CLUSTER_ZONE
```

The output is similar to the following:

```
helloworld-compute1
helloworld-compute2
```

You now have a working cluster!

Slurm Cluster Demo - Cleanup

To delete the cluster run this:

```
gcloud deployment-manager deployments delete slurm
```

Or select all the nodes from the Compute Engine page in the Console and Delete the nodes.

Next Steps

Learning

- Google Getting Started Resources (cloud.google.com/gcp/getting-started)
- Google Quick Starts (cloud.google.com/gcp/getting-started#quick-starts)
- Google Training Courses (cloud.google.com/training)
- Google Cloud Platform Documentation (cloud.google.com/docs)

Resources

- Google Free Tier (cloud.google.com/free)
  - 20 + “free” products and $300 free credit

Cloud Consulting

- CUIT Research Computing Services offers cloud consulting, just contact rcscloud-support@columbia.edu