Google Cloud Guide

Daniel Qian, Ulysses Butler

2024/02/12

1 Setup Education Credits

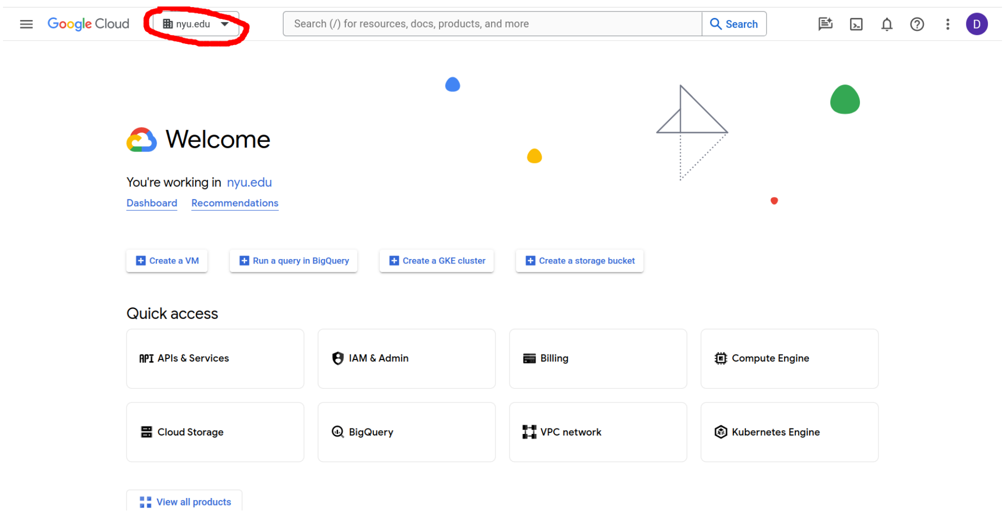
First, claim your Google cloud education credits from the link in this Piazza post. The process should be straightforward, and consists of just verifying your NYU email. Once you are done, you should end up with a "Billing Account for Education" in your account, which contains your $50 of credit.

Please reach out if there are any issues with this process

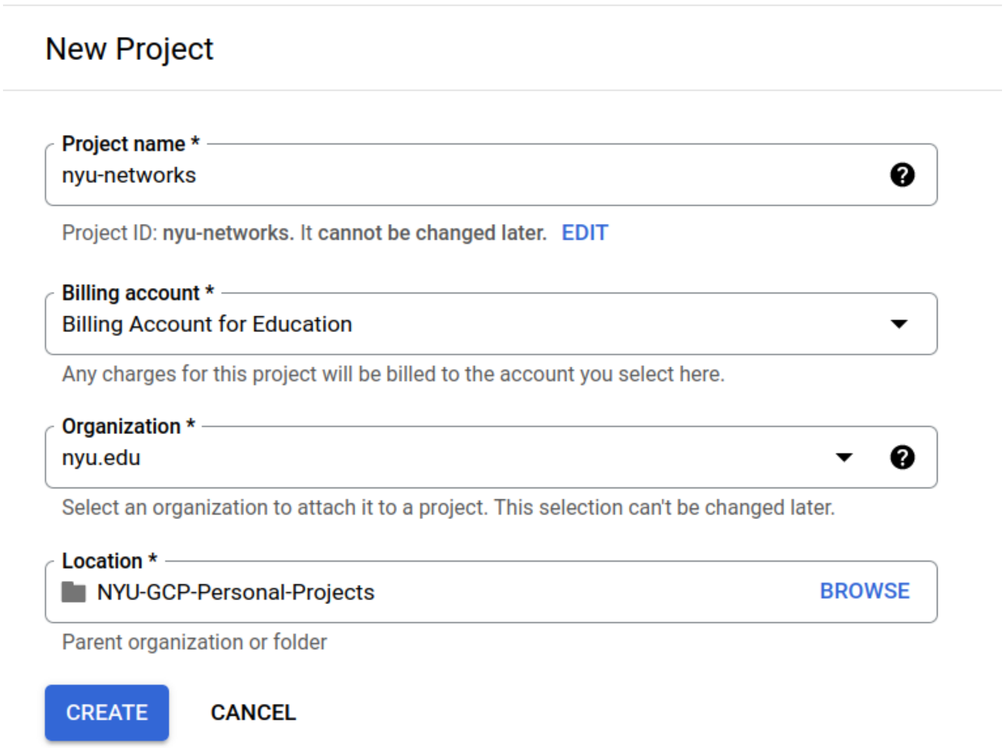
2 Setup project

First, you will need to setup a project to contain your cloud VMs.

This can be done by accessing the dropdown menu in the top left and selecting "New Project in the top right of the modal"



Then, when creating the project, make sure you use the "Billing Account for Education".


New Project

Project name *: nyu-networks

Project ID: nyu-networks. It cannot be changed later. EDIT

Billing account *: Billing Account for Education

Any charges for this project will be billed to the account you select here.

Organization *: nyu.edu

Select an organization to attach it to a project. This selection can't be changed later.

Location *: NYU-GCP-Personal-Projects

Parent organization or folder

You will then need to enable the "Compute Engine", which is Google Cloud's service for provisioning VMs. Click the Hamburger ≡ menu and open the "Compute Engine". You should see the following screen. Enable the compute engine.

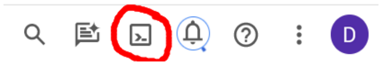
Create the VMs

Once the Compute Engine is enabled (you should see a notification telling you once it is), we can then create the VMs we will use throughout the class. We have provided a set of scripts and templates that will:

- Create 2 VMs, a "server" and "client" (though you can use them however you want)
- Sets up up a "Virtual Private Cloud"[1] that has very loose firewall rules (open TCP and UDP ports) so we can deploy aribitrary network applications. This may not be the best thing to do in practice, but it makes it easier for our purposes.
- Installs all the dependencies you need for the class.

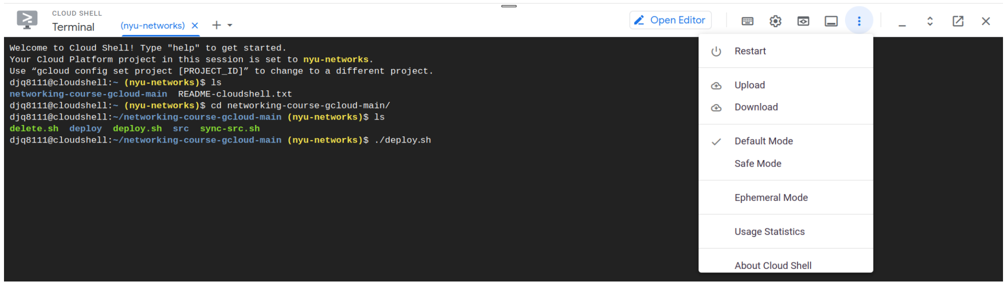
A zip file containing these scripts is available on the course website.

To run these scripts (which use Google Cloud's CLI), we will take advantage of Google's cloud shell, which essentially gives us a free, small VM that has the GCloud CLI tools installed. This is located in the menu bar as shown.



[1]Google's documentation on VPCs if you want to know more. This is a feature implemented by nearly all cloud providers

Open the shell (which is a bash shell) and upload the scripts to the cloud shell in the details dropdown menu as shown

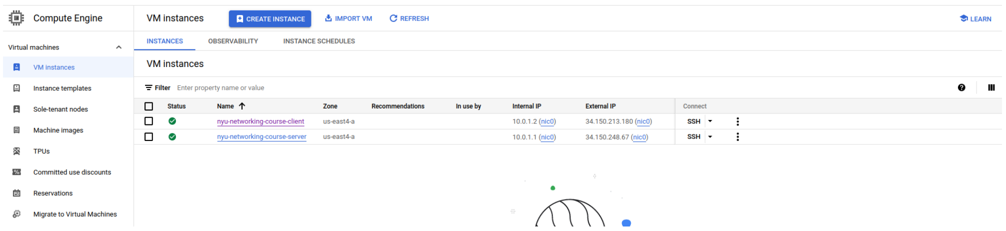


```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to nyu-networks.
Use “gcloud config set project [PROJECT_ID]” to change to a different project.
djq8111@cloudshell:~ (nyu-networks)$ ls
networking-course-gcloud-main  README-cloudshell.txt
djq8111@cloudshell:~ (nyu-networks)$ cd networking-course-gcloud-main/
djq8111@cloudshell:~/networking-course-gcloud-main (nyu-networks)$ ls
delete.sh  deploy  deploy.sh  src  sync-src.sh
djq8111@cloudshell:~/networking-course-gcloud-main (nyu-networks)$ ./deploy.sh
```

Then run the following commands in the directory the `gcloud_scripts.zip` are uploaded to

```
unzip gcloud_scripts.zip
cd scripts
chmod +x deploy.sh
./deploy.sh
```

These commands unzip the archive, modify permissions on the `deploy.sh` script to allow execution, and then run the `deploy.sh` script. You will be asked to confirm and authorize a few things, but after confirming you will see 2 VMs in your Compute Engine dashboard.



Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	nyu-networking-course-client	us-east4-a			10.0.1.2 (nic0)	34.150.213.180 (nic0)	SSH
✓	nyu-networking-course-server	us-east4-a			10.0.1.1 (nic0)	34.150.248.67 (nic0)	SSH

Note, if there are any issues in deployment, you will likely have to delete the deployment with `./delete.sh` to try again.

3 VM Access

The easiest way to access these machines is through the gcloud built in tools. For example, the SSH button console is a relatively easy way to open a shell on your VM, that comes with some nice built in features such as the ability to upload files. Another option is the gcloud ssh command.

3.1 Adding your own ssh keys

Link to the Google Cloud guide

However, you may want to ssh on your own manually, in which case, you would need to add ssh keys to each machine. However, this is not as simple as adding your public key to `authorized_keys`, since Google Cloud manages that file. Instead, we can input our public key on the console, and Google will add that to all the VMs for us!

First create an ssh key (or use an existing one) with

```
ssh-keygen
```

Then, copy the public key (e.g. `id_rsa.pub`) and following the instructions in the above link, copy your public key into the project metadata. This will put your key on all VMs in your project! Note that your key might look like

```
ssh-rsa AAAAB...eRdJsk= dqian@CIMS-PHD-DE3
```

Where the last part is a user describing the user on your personal machine. When entering the key into the UI, change that to the username to the one you want to use on your VM.

3.2 Development/Testing

There are lots of options to edit and deploy your code. Here are a few suggestions, but they aren't the only ones:

- (**recommended**) Create a Github (or other) repo that you clone on both your local machine to edit source code and on the VMs to test them.
- If you setup your own ssh keys above, you can use the VSCode Remote Development Extension, which allows you to work in VSCode on your own machine, but that can access the filesystems and create shells on your VMs.
- Use the provided `sync-src.sh` tool (though this would require installing the GCloud CLI on your machine), which syncs the adjacent src directory with both VMs.
- Just use vim on the VM :)