

# CSE 332: Data Abstractions

## Google Compute Engine & Eclipse

This handout will help you set up Google Compute Engine and Eclipse for the Chess project.

### (1) Setting up Eclipse/Terminal

This project, you will use Google Compute Engine to access more computing power than normal. As such, you will need to ssh to a remote instance. Google Compute Engine lets you do this by clicking a button in the browser, but you may ssh however you like.

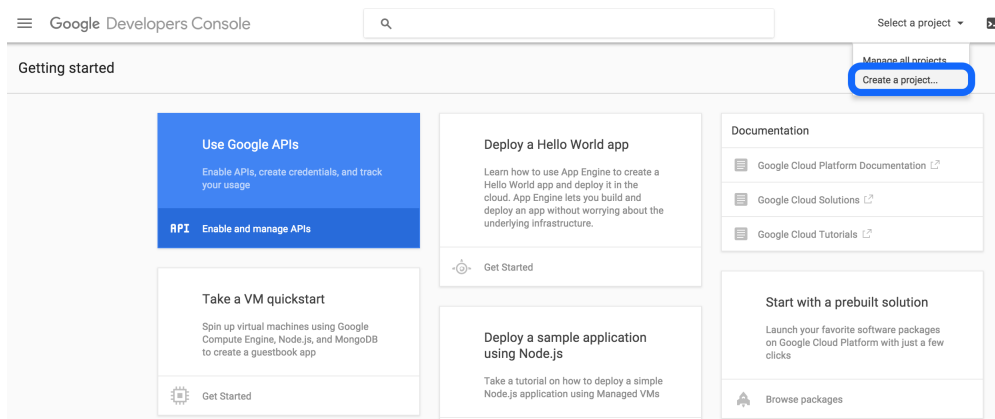
### (2) Setting up Google Compute Engine

Additionally, you will need to create a **project** and **VM Instances** using Google Compute Engine.

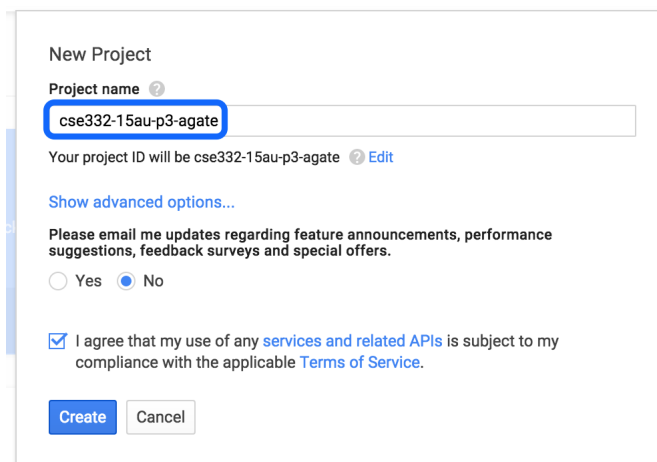
You should begin by going to:

<https://console.developers.google.com/start>

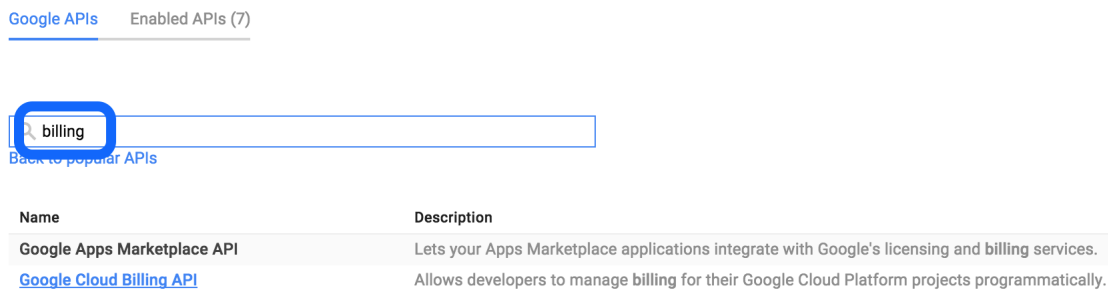
Then, create a project by clicking on the top left:



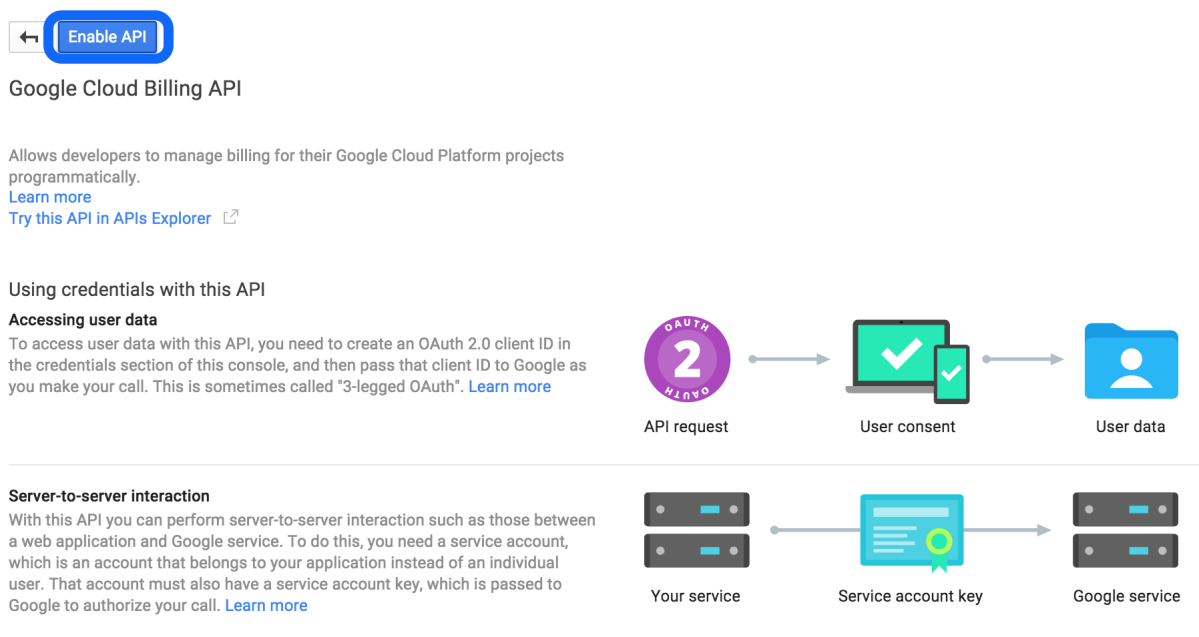
You **must** use `cse332-16wi-p3-<your team name>` as the project name.



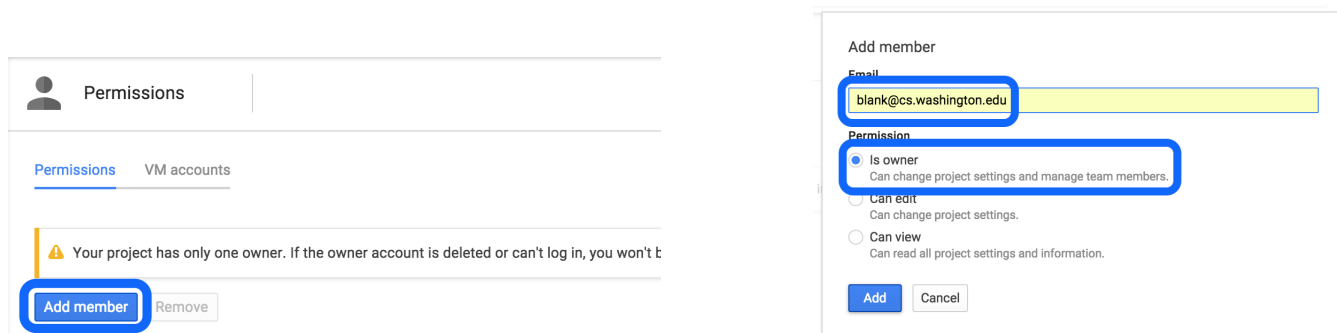
Next, you will enable the Billing API so that you can use the money Google gave us for p3. To do this, click the big “Use Google APIs” button, search for “billing”, and click on “Google Cloud Billing API”:



In the new page that opens up, click on the “Enable API” button:



Next up is adding your partner and Adam to your project. GCE has the distinction between an “owner” and a normal user for a project. You will want to make yourself, your partner, and Adam all owners. To do this, click on the “hamburger menu” on the top left and choose “Permissions”.



Make sure that you use the e-mail address blank@cs.washington.edu and choose the “is owner” box. Note that once you do this step, Adam must *manually* accept each invitation and add the billing account. You should do this sooner rather than later.

### (3) Creating An Instance

Now you should be ready to go with GCE. To test this out, let's try spinning up an instance. Click on VM Instances and click the New Instance button.

When creating an instance, you will have a choice of several options. You *can* choose anything you like, but we recommend the following choices. In particular, if you are creating a "test instance", we recommend you use f1-micro.

- Zone: us-central1-b
- Machine Type:
  - For a test instance: f1-micro
  - For a real instance: n1-highcpu-32
- Image: Debian GNU/Linux 7.8

If you insist on using a Windows Image, you're on your own.

Create a new instance

Name

Zone

Machine type  vCPU: 1, Memory: 0.6 GB

Boot disk  New 10 GB standard persistent disk image

Firewall  Allow HTTP traffic,  Allow HTTPS traffic

#### Select a machine type

Machine types determine the specifications of your machines, such as the amount of memory, virtual cores, and persistent disk limits an instance will have. [Learn more about machine types](#)

##### Shared-core machines

- f1-micro (1 vCPU, 0.6 GB Memory)
- g1-small (1 vCPU, 1.7 GB Memory)

##### Standard machines

- n1-standard-1 (1 vCPU, 3.75 GB Memory)
- n1-standard-2 (2 vCPU, 7.5 GB Memory)
- n1-standard-4 (4 vCPU, 15 GB Memory)
- n1-standard-8 (8 vCPU, 30 GB Memory)
- n1-standard-16 (16 vCPU, 60 GB Memory)
- n1-standard-32 (32 vCPU, 120 GB Memory)

##### High CPU machines

- n1-highcpu-2 (2 vCPU, 1.8 GB Memory)
- n1-highcpu-4 (4 vCPU, 3.6 GB Memory)
- n1-highcpu-8 (8 vCPU, 7.2 GB Memory)
- n1-highcpu-16 (16 vCPU, 14.4 GB Memory)
- n1-highcpu-32 (32 vCPU, 28.8 GB Memory)

##### High memory machines

- 

Once your instance boots, you will be able to ssh into it by clicking the "ssh" button next to it in the list:

<input type="checkbox"/>	Name ^	Zone	Disk	Network	In use by	External IP	Connect
<input checked="" type="checkbox"/>	chessbots	us-central1-f	chessbots	default		104.197.134.107	SSH
<input checked="" type="checkbox"/>	umessage	us-central1-f	umessage	default		104.197.67.89	SSH

#### (4) Connecting To An Instance

Now that you've created your instance, you are ready to `ssh` to it. To do this, click the button in the GCE interface labeled "ssh" in the row of the instance you just created.

You're now connected to your instance!

Once you connect to your instance, you should copy/paste the following command into the terminal:

```
source <(curl -s https://courses.cs.washington.edu/courses/cse332/gce/setup-instance.sh)
```

Make sure you read the instructions that the script prints out. You will need to copy another key to gitlab, and, when you do, if it errors, make sure your key does not have any spaces in it.

#### (5) Running A Bot On A GCE Instance

When you are running one of your bots on a GCE instance, you **should not** use EasyChess, because you do not want to access it with a GUI. Instead, you will use `CloudClient.java`.

In preparation for running your bot on the server, edit the configuration of `CloudClient.java` to be set to your Chess Server Login and the bot you want to run. Then, after you've run the command (above), you should run `CloudClient` on the VM. Finally, to see your bot play the game, you should do the following:

- Run EasyChess on your local machine
- Type "games" to see the list of currently running games
- Type "watch #"

#### (6) GCE Won't Allow Me To Use More Cores!

If you have multiple instances running, the *total* you are allowed is 32 cores. Sometimes, for reasons that I haven't figured out, GCE doesn't let you use 32 cores like it should.

If this happens, follow these instructions and it should start working in a matter of minutes:

- Go to this form:  
<https://docs.google.com/forms/d/1vb2MkAr9JcHrp6myQ3oTxCyBv2c7Iyc5wqIKqE3K4IE/viewform>
- Fill out the required fields at the top, and then the one labeled "Total requested number of cores" in the "US-CENTRAL1 regional quotas" part of the form.
- Then, check back periodically at:  
[https://console.cloud.google.com/compute/quotas?project=YOUR\\_PROJECT](https://console.cloud.google.com/compute/quotas?project=YOUR_PROJECT)  
to see if the 24 goes up to 32.