# CS 3113

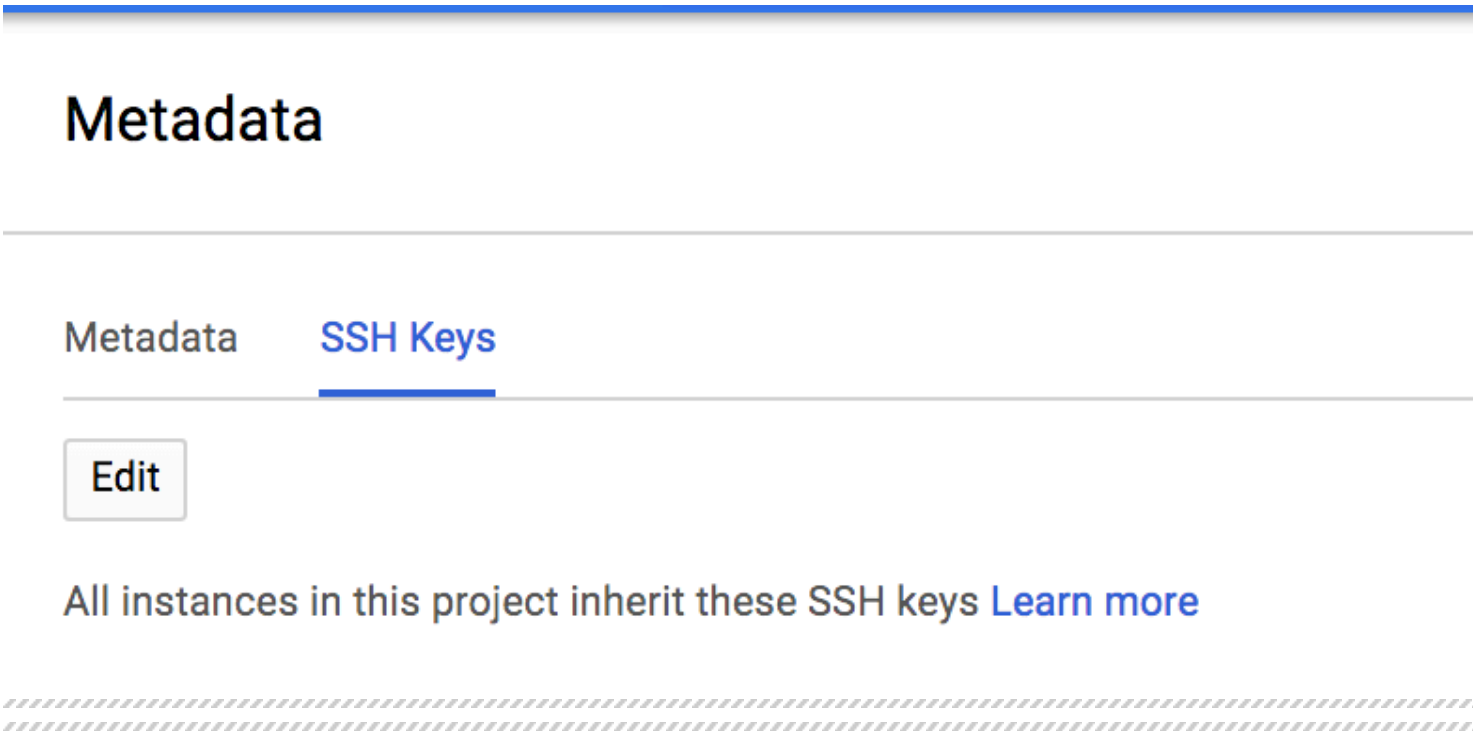
# Homework number 2 (20 Points)

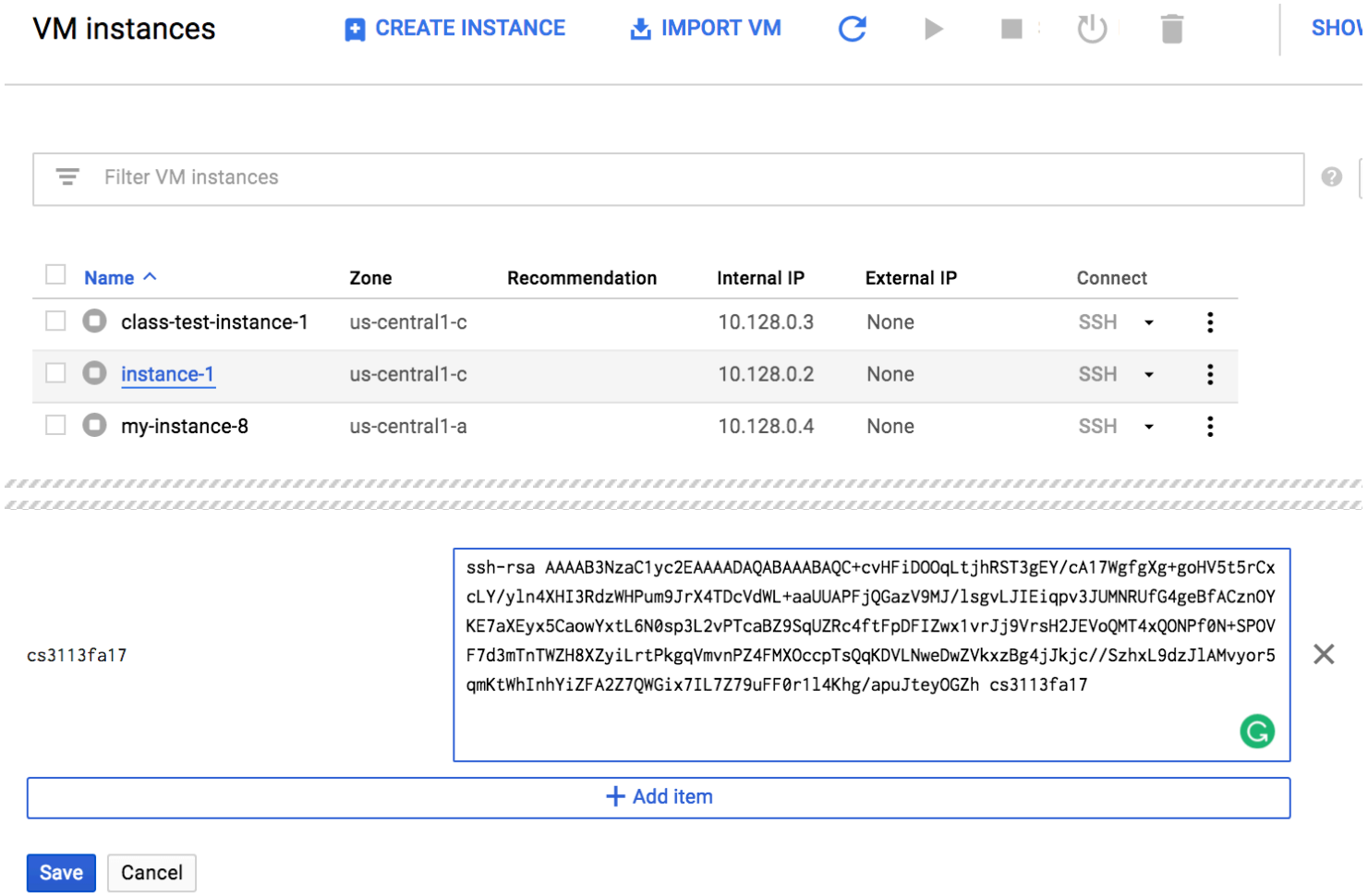
This homework assignment is to prepare you for the C programming in the google cloud environment.

Create f1-micro virtual machine on your google cloud instance. Choose the `us-central1-a` zone. Be sure to select an external ip and to allow all project keys. Use the latest version of Ubuntu.

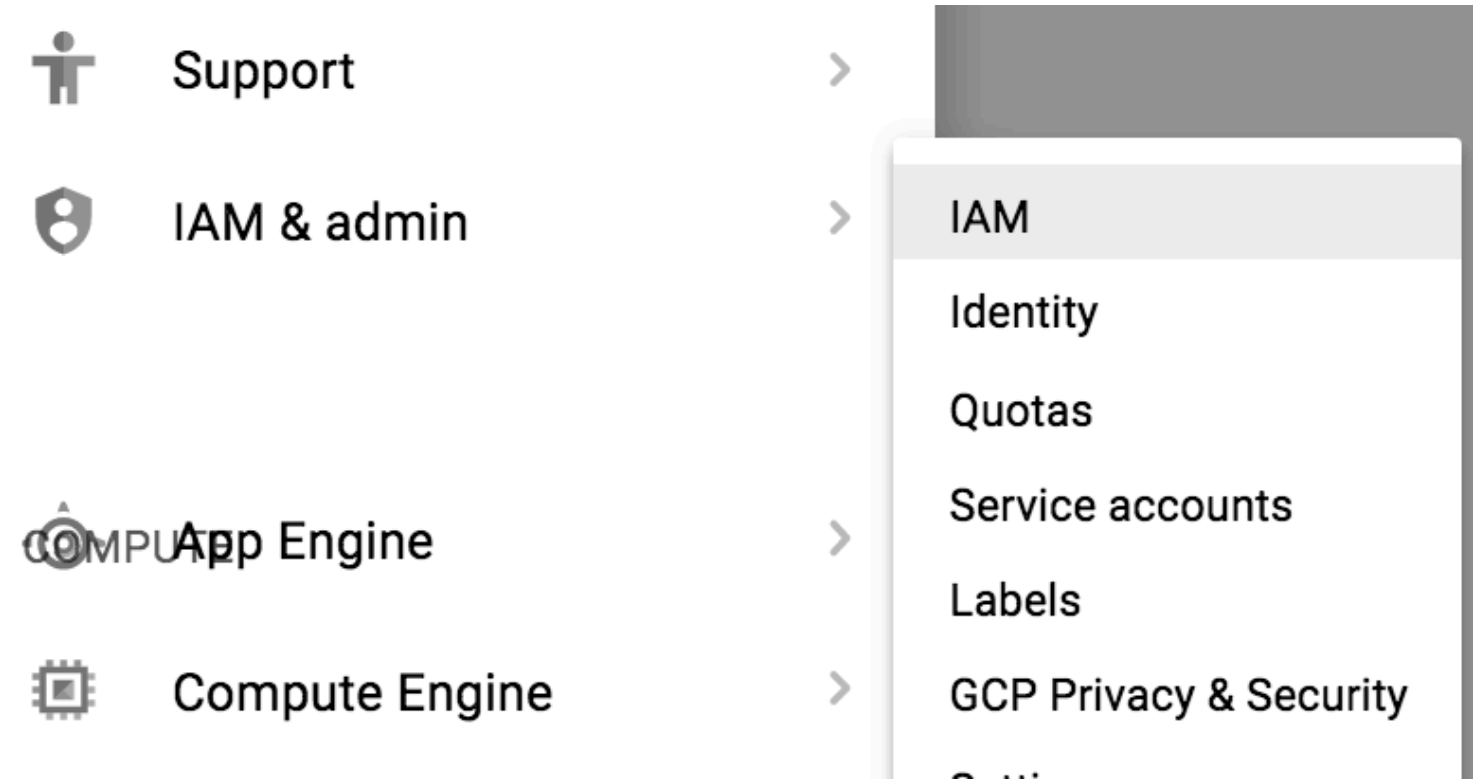
Add the ssh public key for the `cs3113fa17` user. The key can also be downloaded using the command: `wget http://cs.ou.edu/~cgrant/cs3113fa17.pub`
The key is available here: http://cs.ou.edu/~cgrant/cs3113fa17.pub.

To add your key, click on the your instances. Then, select SSH Keys > Edit. Select add item, Enter the Key above, exactly verbatim (there should be no newlines within the key text). Finally, be sure to click save.

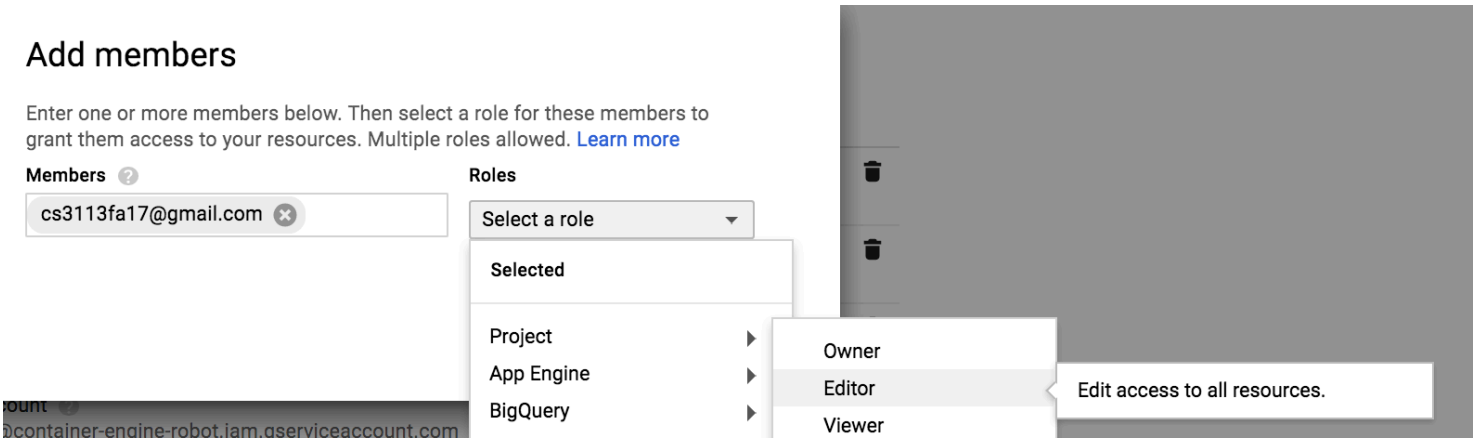





Add the cs3113fa17 user as an editor to the project. First, go to the IAM & admin page for your project. Then, add `cs3113fa17@gmail.com` as an editor for your project. Remember this must be the same project that is used to create your VM.








Run the following start up scripts to update the machine and create new folders.

```bash
# Make a place for your projects
sudo mkdir /projects
sudo chmod 777 /projects
mkdir /projects/hw2/
sudo chmod 777 /projects/hw2

# Install
sudo apt update
sudo apt dist-upgrade
sudo apt install vim htop tmux tree
sudo apt install gcc gdb make
sudo apt install valgrind strace
sudo apt install linux-tools-common linux-tools-generic linux-tools-`uname -r`

cd /projects
wget http://man7.org/tlpi/code/download/tlpi-170610-dist.tar.gz
tar xvzf tlpi-170610-dist.tar.gz
cd tlpi-dist/
make
```

You can run this using a short cut (this practice is generally unsafe): `wget https://www.cs.ou.edu/~cgrant/cs3113fa17.sh`.

Create a c file called `diamond.c` in the `/projects/hw2/` directory. This C-program code when compiled should takes two parameters as command line input (1) an odd integer from 1-15 and (2) a single ASCII character. If the input is some number k, the height and width of the diamond should be (2*k-1). All output should be send to standard out (stdout).

Below is a skeleton for this program:

C

```c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[])
{
    // Naive Error check
    if (argc != 3) {
        fprintf(stderr, "Two parameters an int and a char.\n");
        exit(-1);
    }

    // Get the size of the diamond
    int num = atoi(argv[1]);

    if (num < 1 || num > 15) {
        fprintf(stderr, "Then number must be between 1 and 15");
    }

    char *k;
    // Set the character k
    k = argv[2];

    /** Fill in here **/


    return 0;
}
```

Here is a sample run:

```Bash
barrybonds@instance-1$ gcc -g diamond.c
barrybonds@instance-1$ ./a.out 2 .
 .
...
 .
barrybonds@instance-1$ ./a.out 7 .
       .
      ...
     .....
    .......
   .........
  ...........
.............
  ...........
   .........
    .......
     .....
      ...
       .
```

To test your program we will connect to your created VM. Compile the file using `gcc -g -O3 diamond.c` Then we will execute several test cases and check the output against the expected output. It is important that you give any user permissions to compile and execute your code `chmod -R 644 /projects/hw2/diamond.c`.

You will only need to submit your ip address and zone.

# Grading Criteria:

| Task | Percent |
| --- | --- |
| Instance is reachable | 20% |
| Code compiles | 30% |
| Code Passes all test cases | 50% |
| **Total** | **100%** |