Task 2 - Setting up a vm

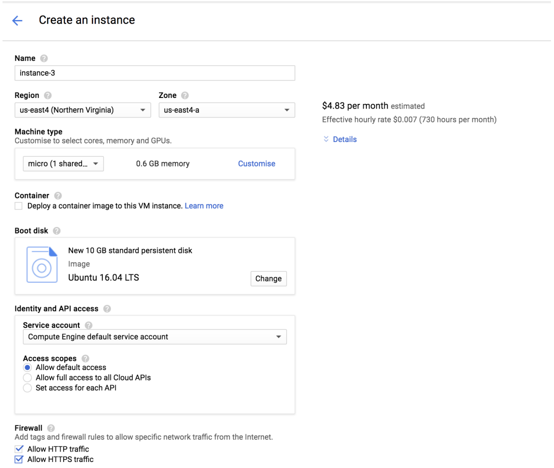
Yuan Fan <yfan@kth.se>   Xuwei Gong<xuweig@kth.se>

I. Creating vm using web console

In this task we will work with Google Cloud Compute Engine, to start with we need to set up with a vm instance, the quickest way is to do on the web console. GCP has provided with several options with vm, but since it's our first try we took the simple configuration.

A. Basic configuration

To launch a virtual instance, go to the Compute Engine menu on the left column of your dashboard and click on VM instances. When you create a VM instance, you need to enter a number of settings for the instance, where ”name” is filled in based on personal preferences; ”region” is the location of the instance machine, as shown in the figure 1 our virtual machine choose the ”europe-west1”; ”machine type” is selected based on personal preferences, In any case, the better the machine performance, the more expensive the price, the use of bonuses will be faster, however to save up the resources we choose the ”micro”; ”startup disk” is to choose to install the system image and adjust the type and size of the disk, the default is Debian 9 system, 10GB mechanical disk, we are more familiar with Ubuntu 16.04 thus it's our option; ”firewall” rule is to release the port 80 and 443 used as a WEB server. Google Cloud has a hardware firewall and needs to manually add firewall rules to its own instance. This is a two-concept approach to the software firewall in the VM instance. We checked both box for further use.



Name: instance-3

Region: us-east4 (Northern Virginia)

Zone: us-east4-a

$4.83 per month estimated

Effective hourly rate $0.007 (730 hours per month)

Machine type

Customise to select cores, memory and GPUs.

micro (1 shared... 0.6 GB memory

Container

□ Deploy a container image to this VM instance.

Boot disk

New 10 GB standard persistent disk

Image

Ubuntu 16.04 LTS

Identity and API access

Service account: Compute Engine default service account

Access scopes

● Allow default access

○ Allow full access to all Cloud APIs

○ Set access for each API

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet.

☑ Allow HTTP traffic

☑ Allow HTTPS traffic

Fig. 1. VM setup configuration

B. SSH Key pair generation

To generate SSH key pair we followed the tutorial in Google documents, the user can manually done it or use a tool like **gcloud** which will prompt you to create them if you don't have SSH keys yet. The command line is shown below:

```
gcloud compute ssh [example-instance]
```

Once the key pair has generated, we can store it in to the Metadata section for further log in.

II. Remote creation of VM

Also there are ways to create vm remotely using gcloud tools or APIs that GCP provided.

A. gcloud tools

Using gcloud is under the situation that the user has gcloud command line tool installed on the host, there's also a tutorial for it. Then the simple example command line are shown:

```
gcloud compute instances create
[INSTANCE_NAME]
--image [IMAGE_ID]
--image-project [IMAGE_PROJECT]
```

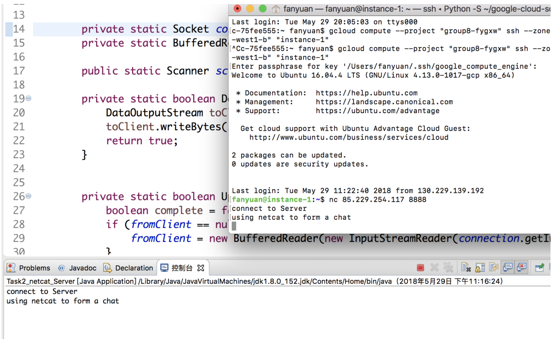
where the parameters need to be defined such as your specific image, the image project to which that image belongs and so on.

B. API Request

The API library is also a useful tool to start a new instance or create a vm. By handling the calling of REST API and construct a POST request with instance URI, you can create a VM instance by using the **initializeParam** property for each additional disk, where the property need to be claimed in the body of code. Apart from the parameters mentioned in the gcloud tools, the initial parameters also need to include the disk type, size and the zone of your instance in your request. And to start an instance in the API, you need to send a request with *Source image,Network interface, Machine type.*

III. Remote Access the VM

In order to remote access the vm, we wrote a simple code using Java to realize the Server side of netcat tools. As we using ssh to login to the vm, once the netcat tool is installed in the vm it can perform a simple netcat Client side. In our TCP implementation, we use *Socket, ServerSocket* to bind with the port 8888, we put server on listen mode and conduct the connection using `"nc hostIP port"` command. Once the Server and Client can communicate with each other through the socket the host from our side can access vm and can also perform the file transmission. Figure 2 shows the simple test of TCP:



```
    private static Socket co
    private static BufferedR

    public static Scanner sc

    private static boolean D
        DataOutputStream toC
        toClient.writeBytes(
        return true;
    }


    private static boolean U
        boolean complete = f
        if (fromClient == nu
            fromClient = new BufferedReader(new InputStreamReader(connection.getI
        }
```

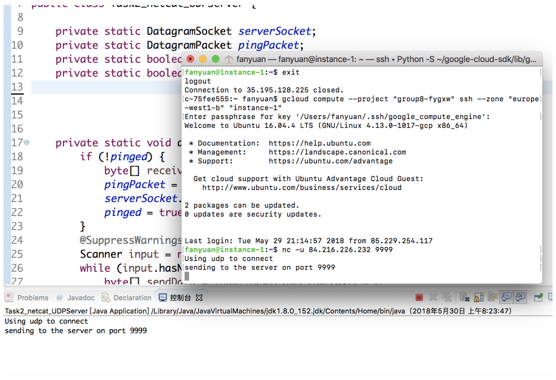
fanyuan — fanyuan@instance-1: ~ — ssh ◂ Python -S ~/google-cloud-s

```
Last login: Tue May 29 20:05:03 on ttys000
c-75fee555:~ fanyuan$ gcloud compute --project "group8-fygxw" ssh --zone
-west1-b" "instance-1"
^Cc-75fee555:~ fanyuan$ gcloud compute --project "group8-fygxw" ssh --zo
-west1-b" "instance-1"
Enter passphrase for key '/Users/fanyuan/.ssh/google_compute_engine':
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.13.0-1017-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

2 packages can be updated.
0 updates are security updates.


Last login: Tue May 29 11:22:40 2018 from 130.229.139.192
fanyuan@instance-1:~$ nc 85.229.254.117 8888
connect to Server
using netcat to form a chat
```

```
Task2_netcat_Server [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/bin/java（2018年5月29日 下午11:16:24）
connect to Server
using netcat to form a chat
```

Fig. 2. Access through TCP

Similarly with UDP, in the test we bind port 9999 as an example.



```
    private static DatagramSocket serverSocket;
    private static DatagramPacket pingPacket;
    private static booleo
    private static booleo



    private static void c
        if (!pinged) {
            byte[] receiv
            pingPacket =
            serverSocket.
            pinged = true
        }
        @SuppressWarnings
        Scanner input = r
        while (input.hasN
            byte[] sendDo
```

fanyuan — fanyuan@instance-1: ~ — ssh ◂ Python -S ~/google-cloud-sdk/lib/g...

```
fanyuan@instance-1:~$ exit
logout
Connection to 35.195.128.225 closed.
c-75fee555:~ fanyuan$ gcloud compute --project "group8-fygxw" ssh --zone "europe
-west1-b" "instance-1"
Enter passphrase for key '/Users/fanyuan/.ssh/google_compute_engine':
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.13.0-1017-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

2 packages can be updated.
0 updates are security updates.


Last login: Tue May 29 21:14:57 2018 from 85.229.254.117
fanyuan@instance-1:~$ nc -u 84.216.226.232 9999
Using udp to connect
sending to the server on port 9999
```

```
Task2_netcat_UDPServer [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_152.jdk/Contents/Home/bin/java（2018年5月30日 上午8:23:47）
Using udp to connect
sending to the server on port 9999
```

Fig. 3. Access through UDP