STEP BY STEP INSTRUCTIONS FOR BUILDING A DEEP LEARNING MODEL TO DETECT TABLES FROM IMAGES

INTRODUCTION

This how to guide is inspired by the blog post written by Christian Beckmann (https://blog.goodaudience.com/table-detection-using-deep-learning-7182918d778). This is an excellent blog post which explains how to implement a table detection system

From the comments section of the above blog post and by receiving several requests from data science beginners, I realised many beginners are having trouble with building this system because of some minor environment or version issues.
So I decided to write this guide which will provide step by step instructions on how to build a deep learning Model to detect tables from images.

At the end of this article I have also provided a small program that can be used to predict the tables using the model we have built. You can refer to this if you would like to integrate this model into your own python applications.

If you follow all the 84 steps as specified here you will end up with a deep learning system that can identify a table from an image and give you X,Y coordinates for the same.

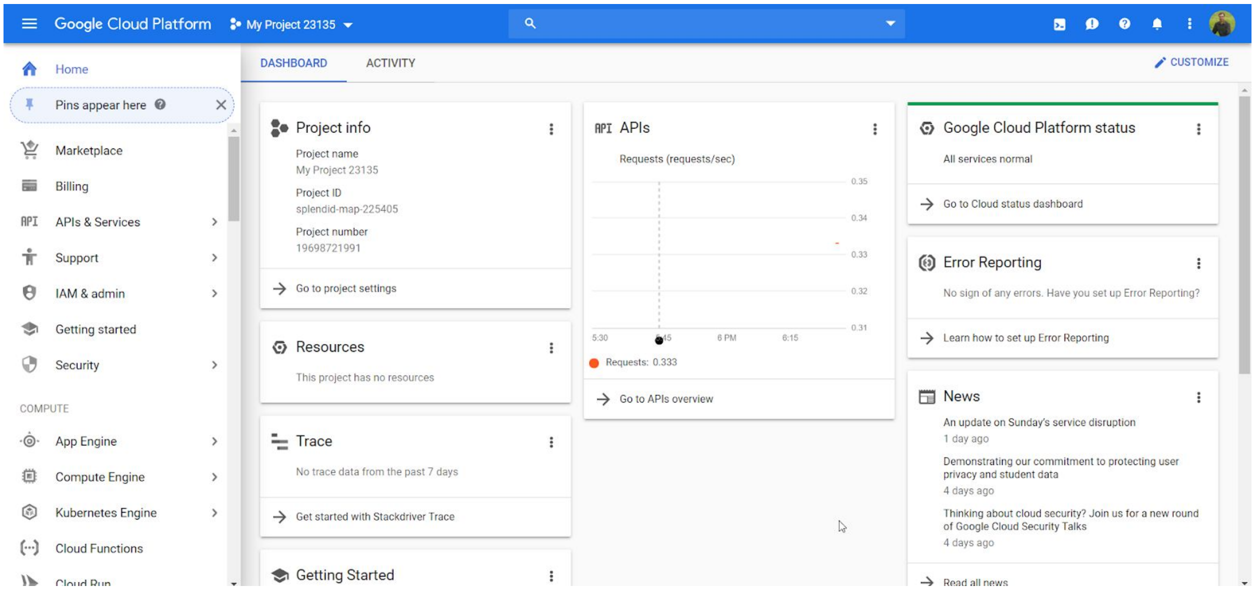
SETTING UP ENVIRONMENT

SInce we are going to build a deep learning model, we need a system with GPU support. I have chosen Google Cloud for the same.

CAUTION : As of today, writing this blog, Google Cloud has been providing $300 free credit for 12 months. You still need to provide your payment details. **Please make sure to read the complete details of this free tier program to understand how it works and the costs involved. (https://cloud.google.com/free/docs/gcp-free-tier)**

1) Visit http://cloud.google.com

2) Click on the "Get started for free” button
3) Sign in with your gmail credentials
4) You need to provide your payment details
5) Once done, will be redirected to your Google Cloud console,



Project name
My Project 23135
Project ID
splendid-map-225405
Project number
19698721991

Requests: 0.333

All services normal

No sign of any errors. Have you set up Error Reporting?

This project has no resources

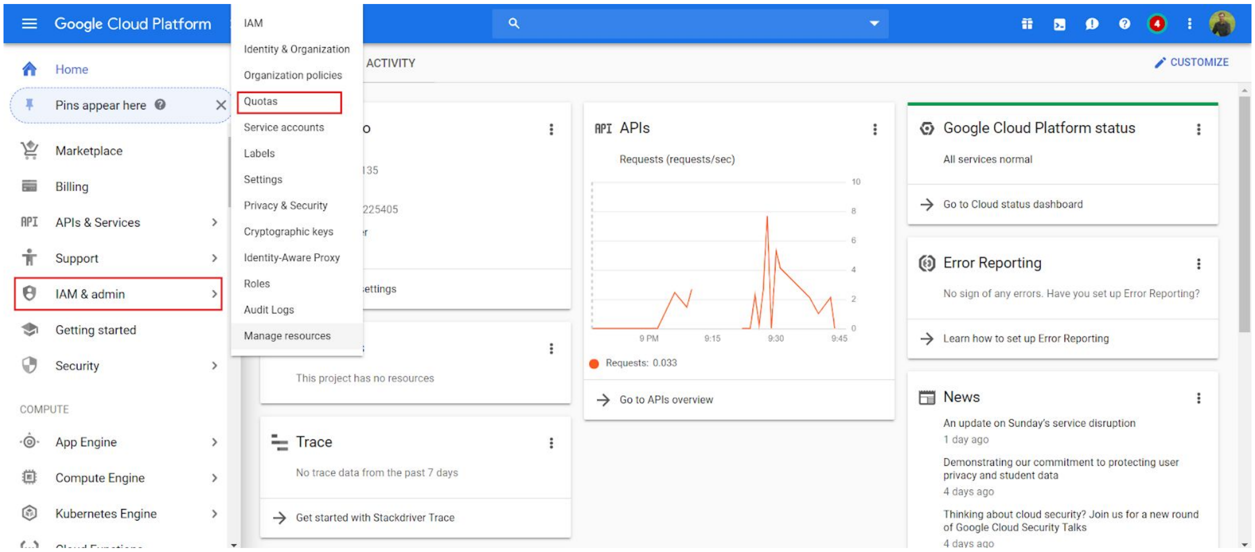
No trace data from the past 7 days

An update on Sunday's service disruption
1 day ago

Demonstrating our commitment to protecting user privacy and student data
4 days ago

Thinking about cloud security? Join us for a new round of Google Cloud Security Talks
4 days ago

6) Select “IAM & admin” and then “Quotas” from the menu.



Requests: 0.033

All services normal

No sign of any errors. Have you set up Error Reporting?

This project has no resources

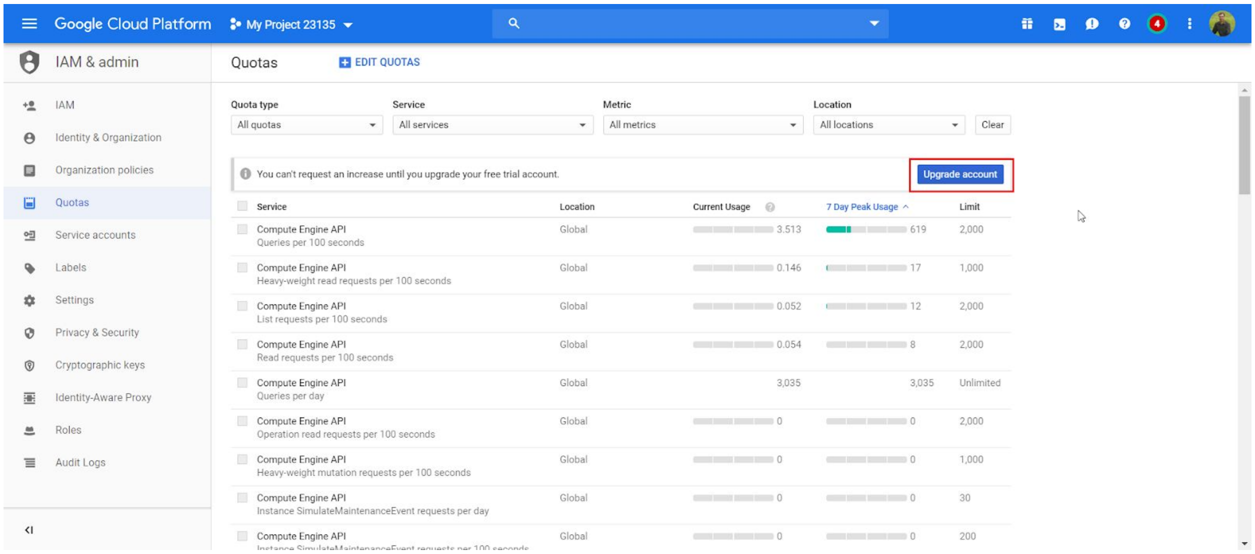
No trace data from the past 7 days

An update on Sunday's service disruption
1 day ago

Demonstrating our commitment to protecting user privacy and student data
4 days ago

Thinking about cloud security? Join us for a new round of Google Cloud Security Talks
4 days ago

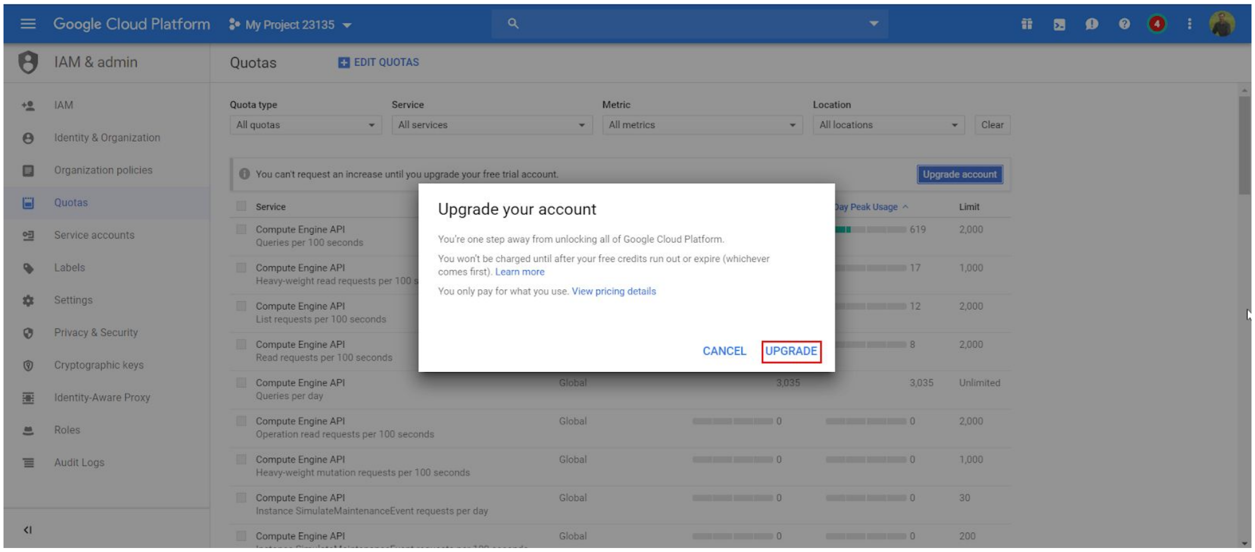
7) Click on the “Upgrade account” button and the “UPGRADE”



Quota type: All quotas
Service: All services
Metric: All metrics
Location: All locations

You can't request an increase until you upgrade your free trial account.

Service	Location	Current Usage	7 Day Peak Usage	Limit
Compute Engine API Queries per 100 seconds	Global	3.513	619	2,000
Compute Engine API Heavy-weight read requests per 100 seconds	Global	0.146	17	1,000
Compute Engine API List requests per 100 seconds	Global	0.052	12	2,000
Compute Engine API Read requests per 100 seconds	Global	0.054	8	2,000
Compute Engine API Queries per day	Global	3,035	3,035	Unlimited
Compute Engine API Operation read requests per 100 seconds	Global	0	0	2,000
Compute Engine API Heavy-weight mutation requests per 100 seconds	Global	0	0	1,000
Compute Engine API Instance SimulateMaintenanceEvent requests per day	Global	0	0	30
Compute Engine API	Global	0	0	200



Upgrade your account

You're one step away from unlocking all of Google Cloud Platform.

You won't be charged until after your free credits run out or expire (whichever comes first). Learn more

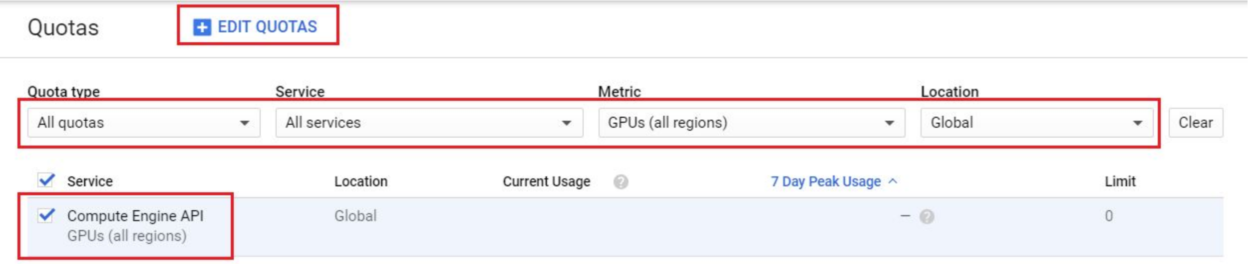
You only pay for what you use. View pricing details

8) After upgrading your account again, select “IAM & admin” and then “Quotas” from the menu.

9) Filter the quota with below parameters (as shown in below image)

Quota type = All quotas
Service = All services
Metric = GPUs (all regions)
Location = Global

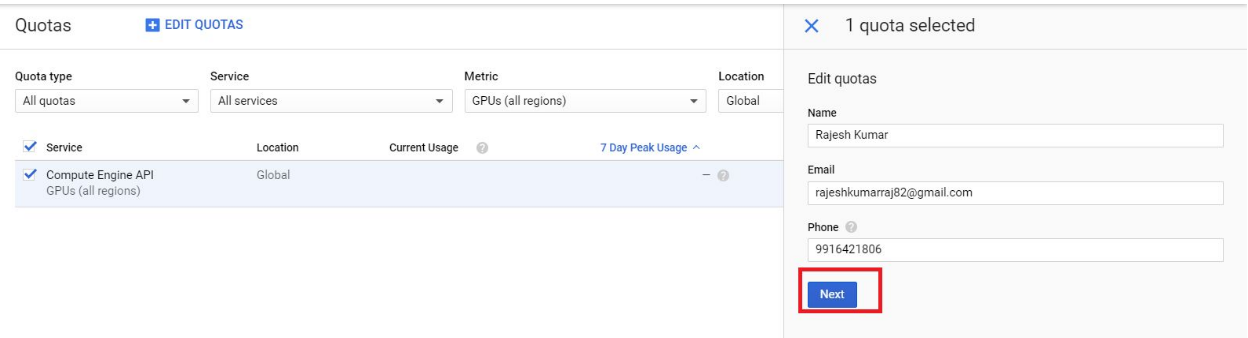
10) Select the quota by clicking on the checkbox and click on the "EDIT QUOTAS” link



Quota type: All quotas
Service: All services
Metric: GPUs (all regions)
Location: Global

☑ Service	Location	Current Usage	7 Day Peak Usage	Limit
☑ Compute Engine API GPUs (all regions)	Global		–	0

11) Fill the details as shown below and click on the "Next" button,

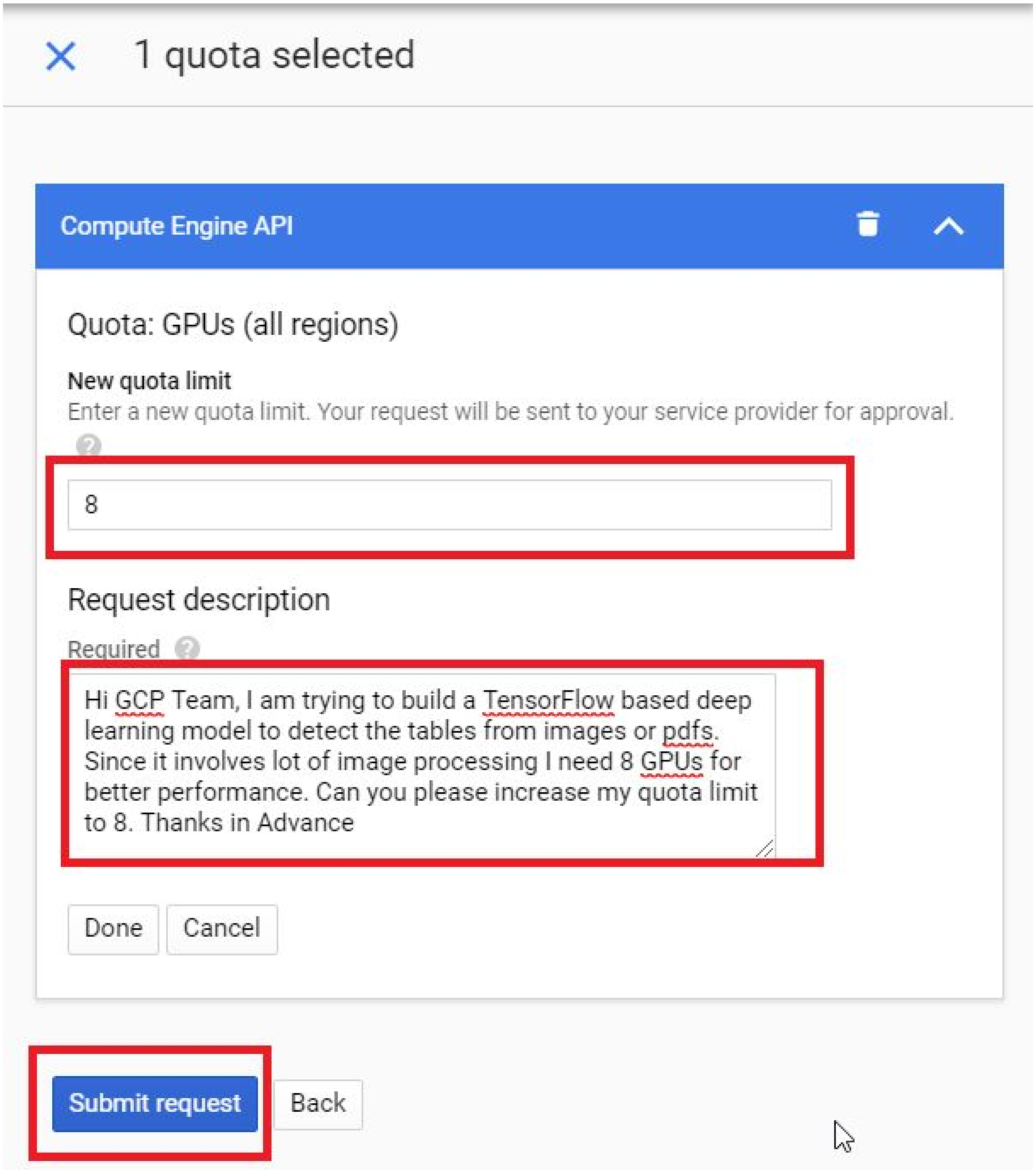


Quota type: All quotas
Service: All services
Metric: GPUs (all regions)
Location: Global

☑ Service	Location	Current Usage	7 Day Peak Usage
☑ Compute Engine API GPUs (all regions)	Global		–

Name: Rajesh Kumar
Email: rajeshkumarraj82@gmail.com
Phone: 9916421806

12) Specify the “New quota limit” as 8 and specify a “Request description” as shown below. Click on “Submit request” button.



1 quota selected

Compute Engine API

Quota: GPUs (all regions)

New quota limit

Enter a new quota limit. Your request will be sent to your service provider for approval.

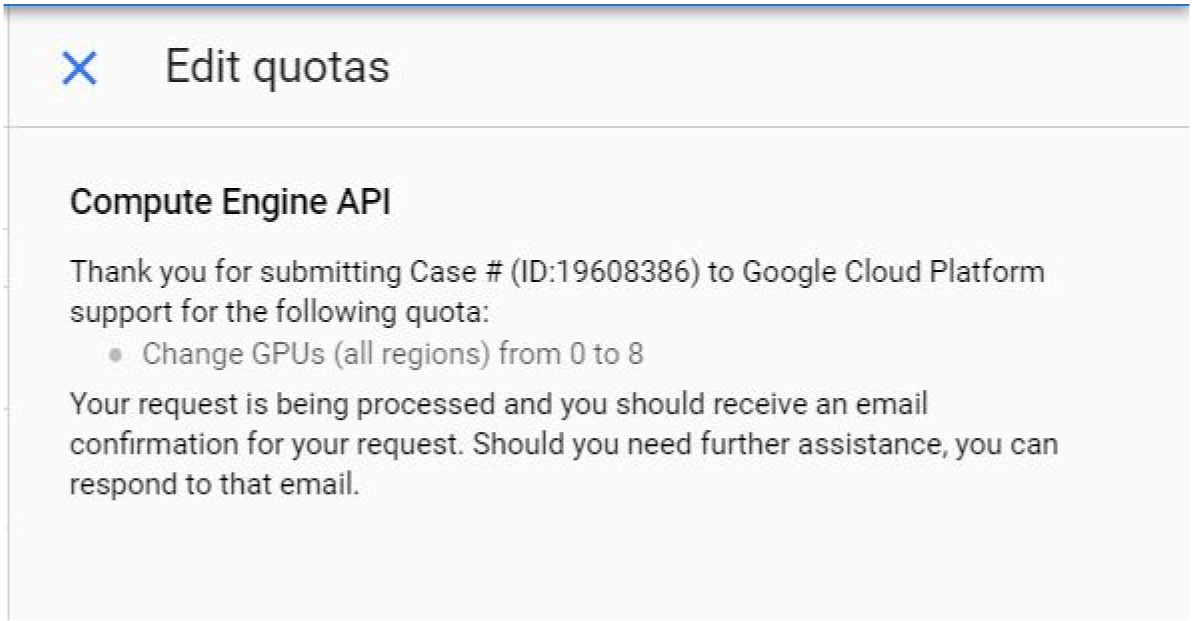
8

Request description

Required

Hi GCP Team, I am trying to build a TensorFlow based deep learning model to detect the tables from images or pdfs. Since it involves lot of image processing I need 8 GPUs for better performance. Can you please increase my quota limit to 8. Thanks in Advance

13) You will get a confirmation as below. Now wait for the email for the confirmation. (Google says it may take 2 business days to process. I got it in about two hours)



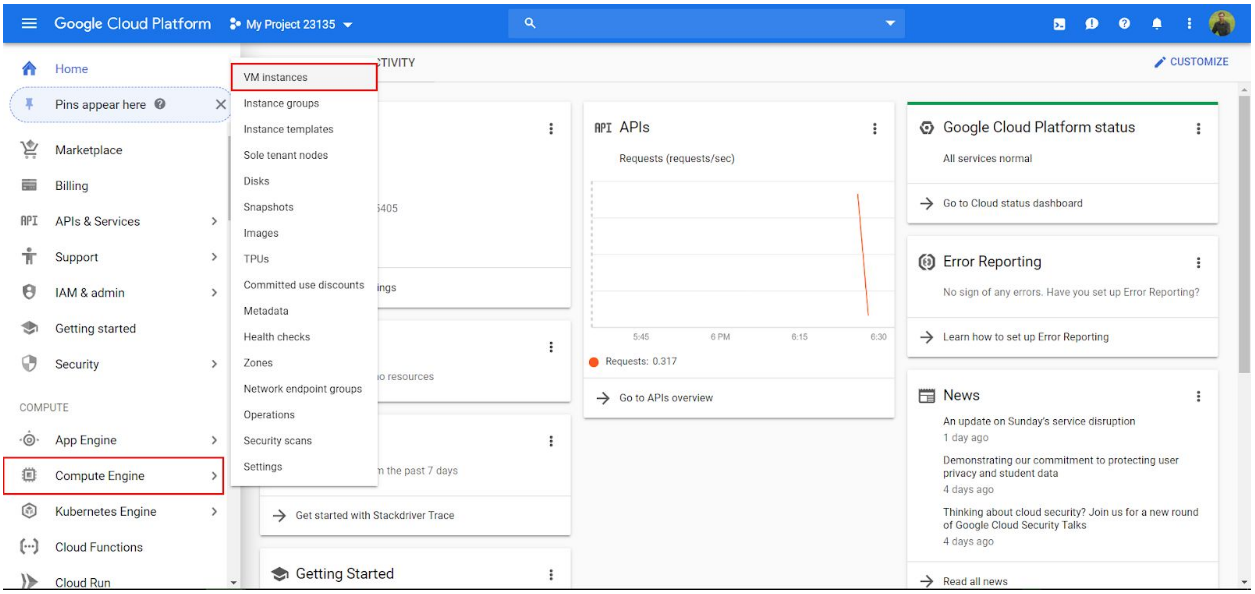
Compute Engine API

Thank you for submitting Case # (ID:19608386) to Google Cloud Platform support for the following quota:

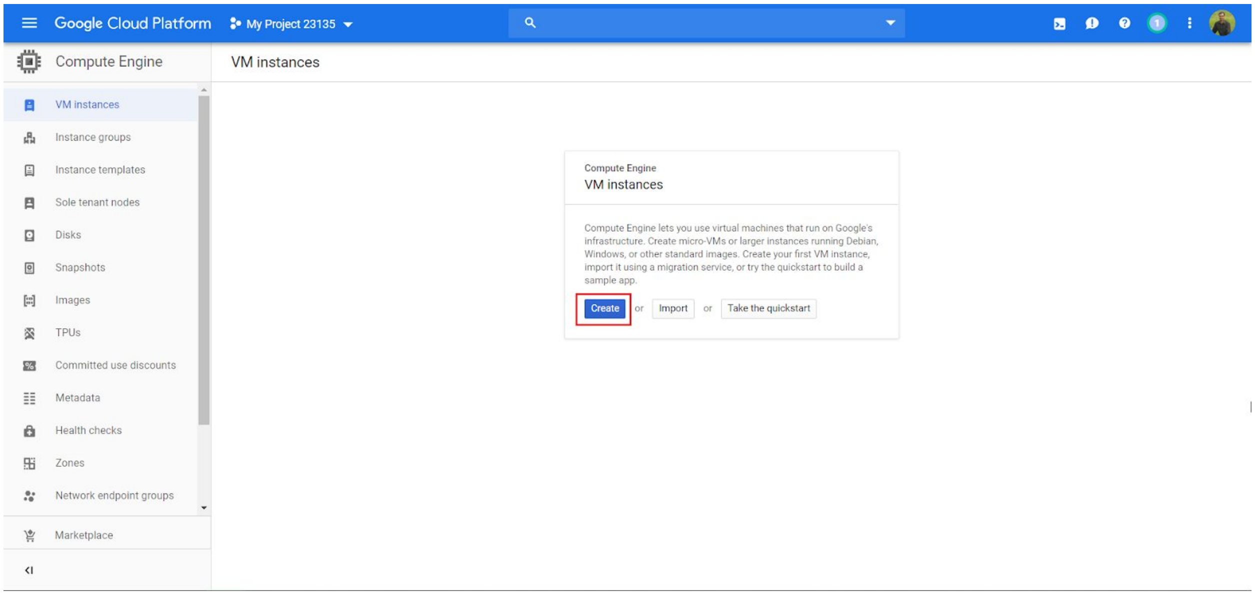
- Change GPUs (all regions) from 0 to 8

Your request is being processed and you should receive an email confirmation for your request. Should you need further assistance, you can respond to that email.

14) Once you receive the approval email from google then proceed with the next steps. Select "Compute Engine" and then "VM instances" from the menu.



15) Click on "Create" button

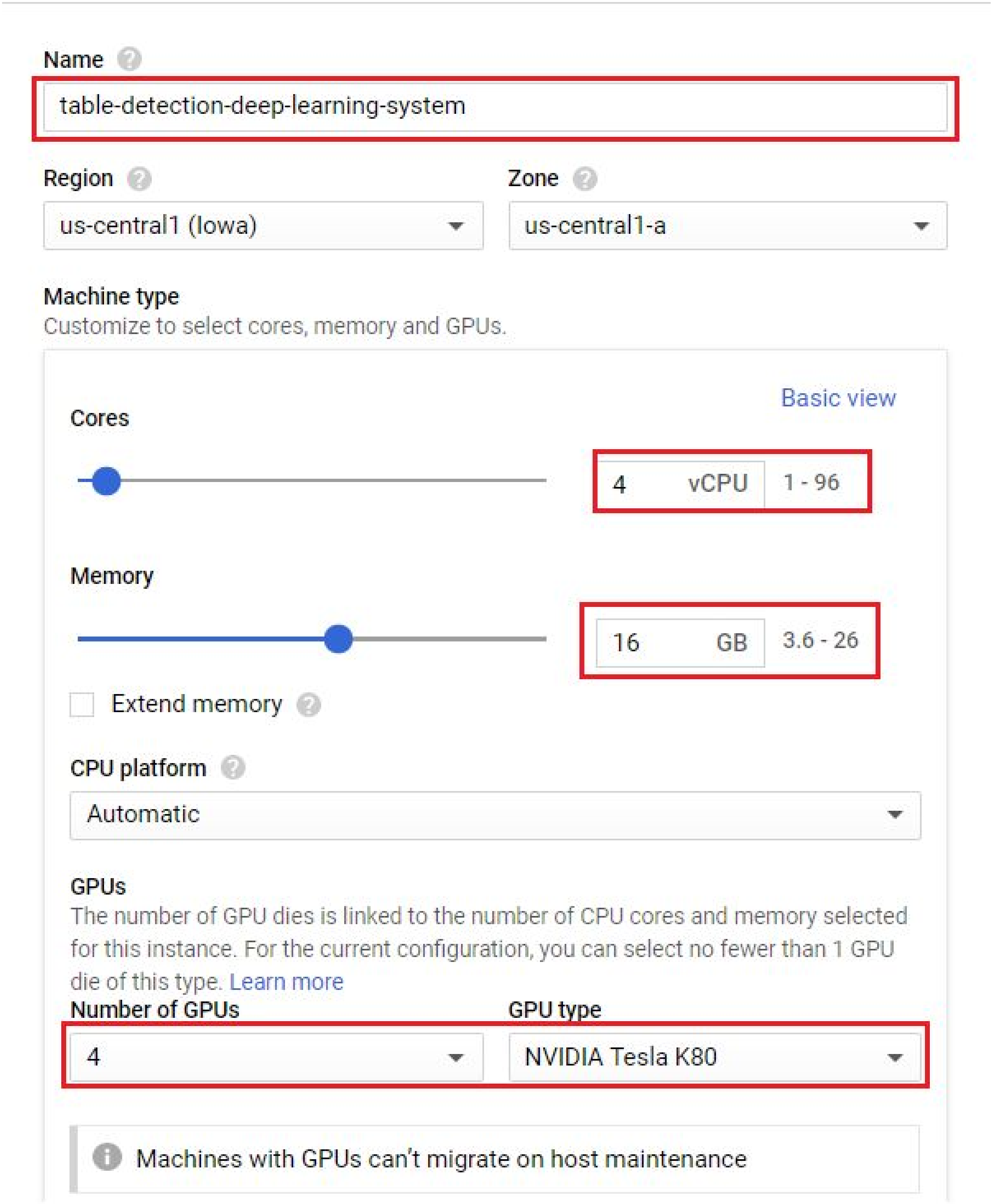


Compute Engine
VM instances

Compute Engine lets you use virtual machines that run on Google's infrastructure. Create micro-VMs or larger instances running Debian, Windows, or other standard images. Create your first VM instance, import it using a migration service, or try the quickstart to build a sample app.

16) “Create an instance” form will be opened.
17) Enter “Name” as “table-detection-deep-learning-system”
18) Click on “Customize” link under “Machine type”
19) Select 4 Core CPUs and 16GB Memory.
20) Specify “Number of GPUs” as 4 and “GPU type” as “NVIDIA Tesla K80”
21) Select “Boot disk” as “Ubuntu 16.04 LTS” specify disk size as 20 GB
22) Click on “Create” button.

Note: Sometimes Google may ask you to try a different zone or try later if there are no resources available. So you can try selecting a different zone.



Name

table-detection-deep-learning-system

Region: us-central1 (Iowa)

Zone: us-central1-a

Machine type

Customize to select cores, memory and GPUs.

Cores: 4 vCPU 1 - 96

Memory: 16 GB 3.6 - 26

□ Extend memory

CPU platform: Automatic

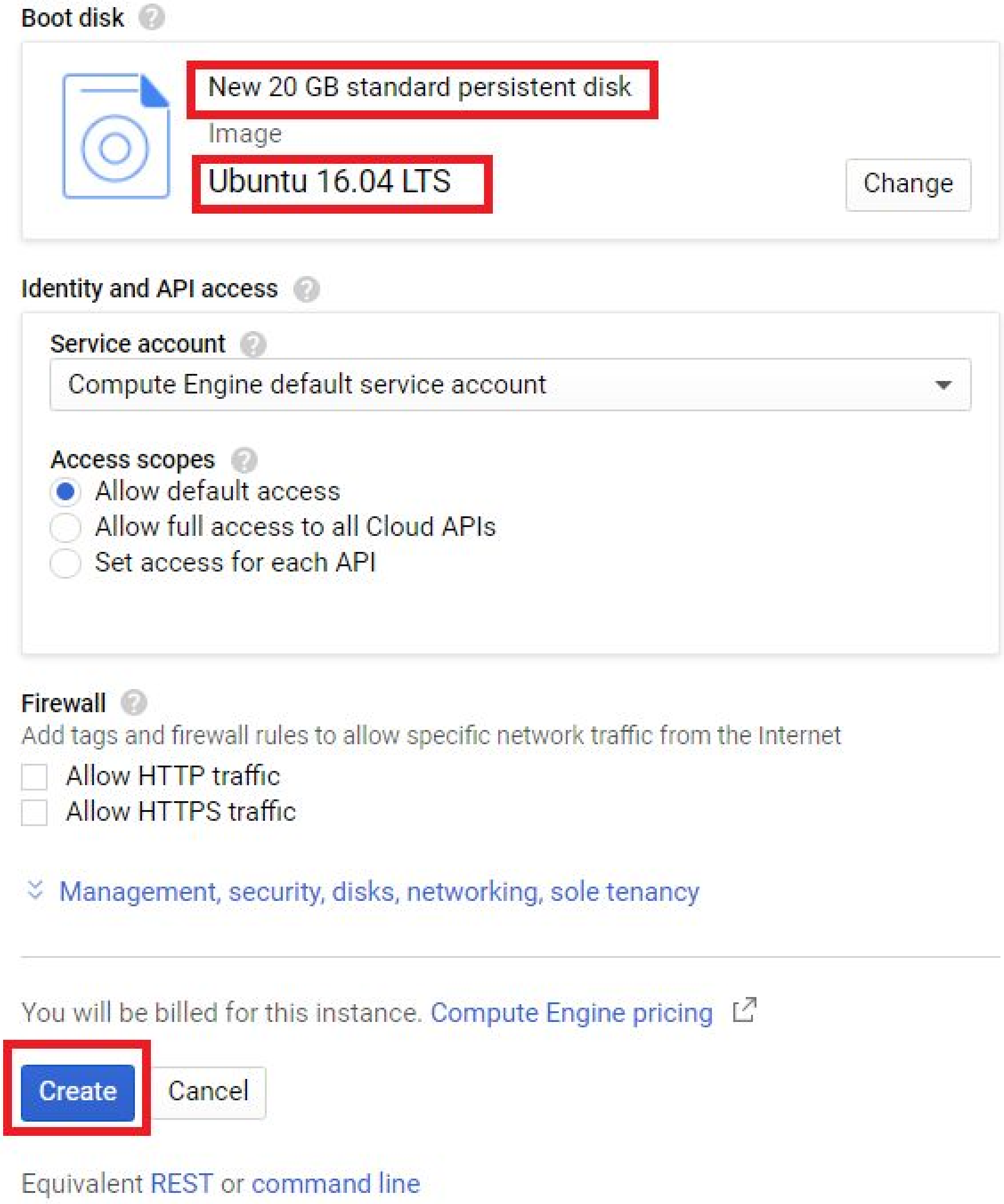
GPUs

The number of GPU dies is linked to the number of CPU cores and memory selected for this instance. For the current configuration, you can select no fewer than 1 GPU die of this type. Learn more

Number of GPUs: 4

GPU type: NVIDIA Tesla K80

Machines with GPUs can't migrate on host maintenance



Boot disk

New 20 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

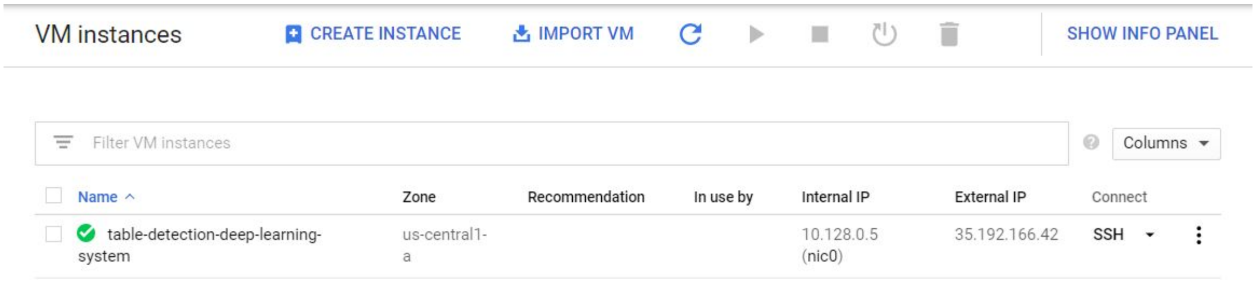
□ Allow HTTP traffic

□ Allow HTTPS traffic

You will be billed for this instance. Compute Engine pricing

Equivalent REST or command line

23) The VM instance will be created as shown below,



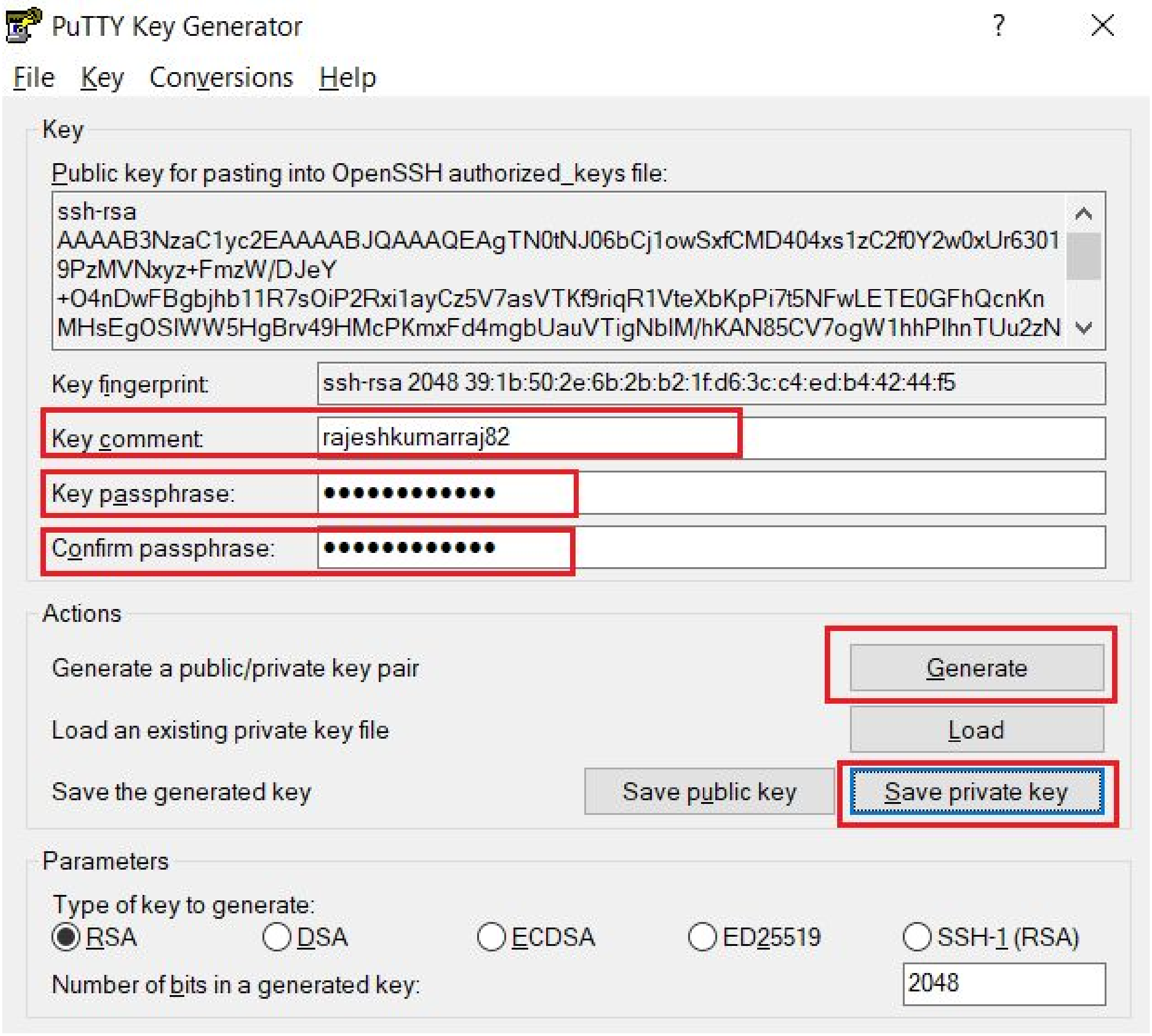
Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
table-detection-deep-learning-system	us-central1-a			10.128.0.5 (nic0)	35.192.166.42	SSH

CONNECTING TO THE GOOGLE CLOUD VM USING PUTTY

Please follow the instructions below to connect to the VM instance we created above.

24) Download putty.exe and puttygen.exe from the below URLs,
http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe
http://the.earth.li/~sgtatham/putty/latest/x86/puttygen.exe

25) Execute puttygen.exe
26) Click on “Generate” button and keep moving the cursor as instructed.
27) Enter your desired username under “Key comment”
28) Enter your desired password under “Key passphrase”
29) Click on “Save private key” button and save the private key file as “private_key.ppk” in your desired location



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAgTN0tNJ06bCj1owSxfCMD404xs1zC2f0Y2w0xUr6301
9PzMVNxyz+FmzW/DJeY
+O4nDwFBgbjhb11R7sOiP2Rxi1ayCz5V7asVTKf9riqR1VteXbKpPi7t5NFwLETE0GFhQcnKn
MHsEgOSIWW5HgBrv49HMcPKmxFd4mgbUauVTigNbIM/hKAN85CV7ogW1hhPIhnTUu2zN
```

Key fingerprint: ssh-rsa 2048 39:1b:50:2e:6b:2b:b2:1f:d6:3c:c4:ed:b4:42:44:f5

Key comment: rajeshkumarraj82

Key passphrase: •••••••••••••

Confirm passphrase: •••••••••••••

Generate a public/private key pair

Load an existing private key file

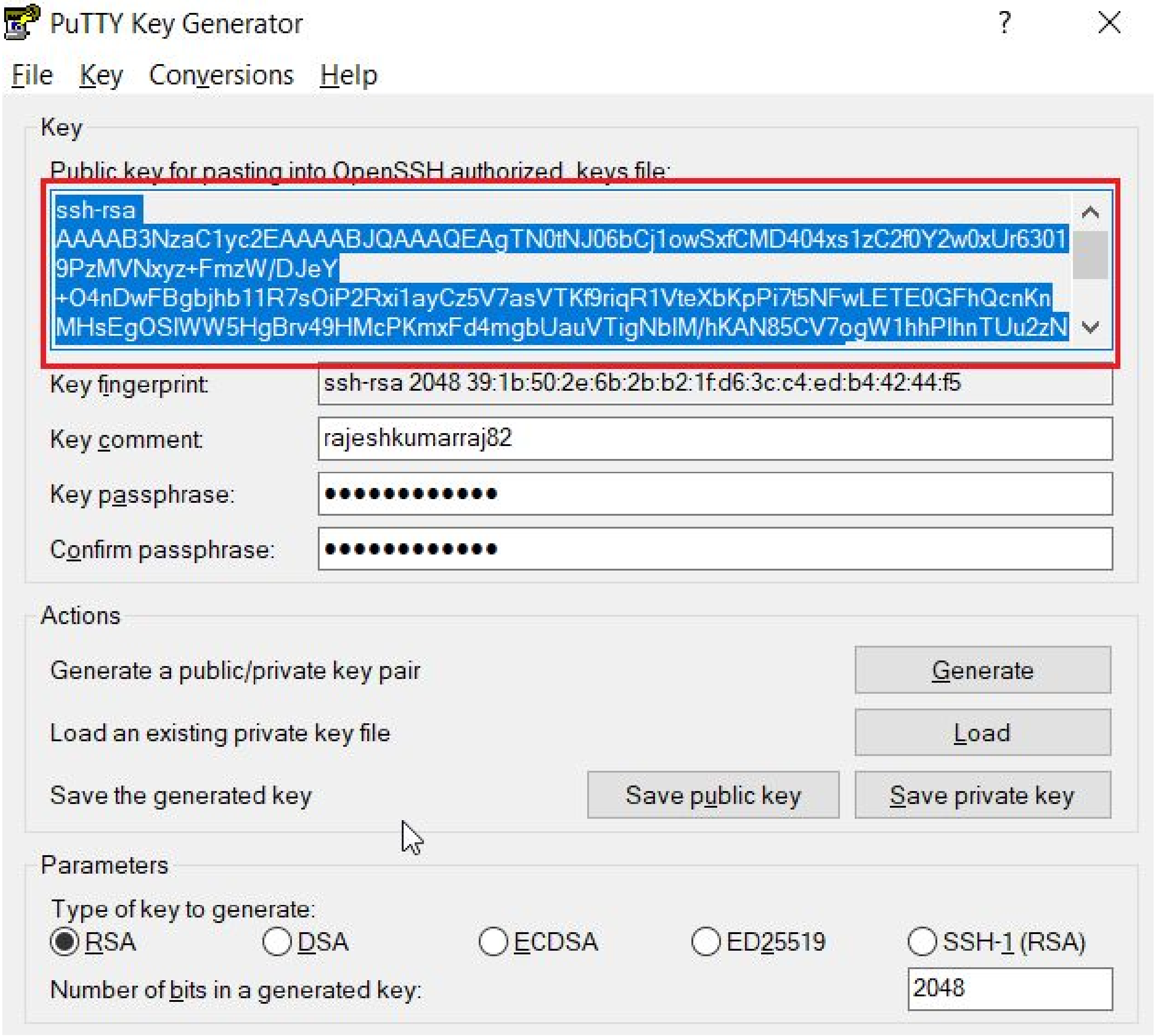
Save the generated key

Type of key to generate:

● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

30) Copy all the public key text highlighted below (You should copy the whole text as it is)



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAgTN0tNJ06bCj1owSxfCMD404xs1zC2f0Y2w0xUr6301
9PzMVNxyz+FmzW/DJeY
+O4nDwFBgbjhb11R7sOiP2Rxi1ayCz5V7asVTKf9riqR1VteXbKpPi7t5NFwLETE0GFhQcnKn
MHsEgOSIWW5HgBrv49HMcPKmxFd4mgbUauVTigNbIM/hKAN85CV7ogW1hhPlhnTUu2zN
```

Key fingerprint: ssh-rsa 2048 39:1b:50:2e:6b:2b:b2:1f:d6:3c:c4:ed:b4:42:44:f5

Key comment: rajeshkumarraj82

Key passphrase: ●●●●●●●●●●●●●

Confirm passphrase: ●●●●●●●●●●●●●

Generate a public/private key pair

Load an existing private key file

Save the generated key

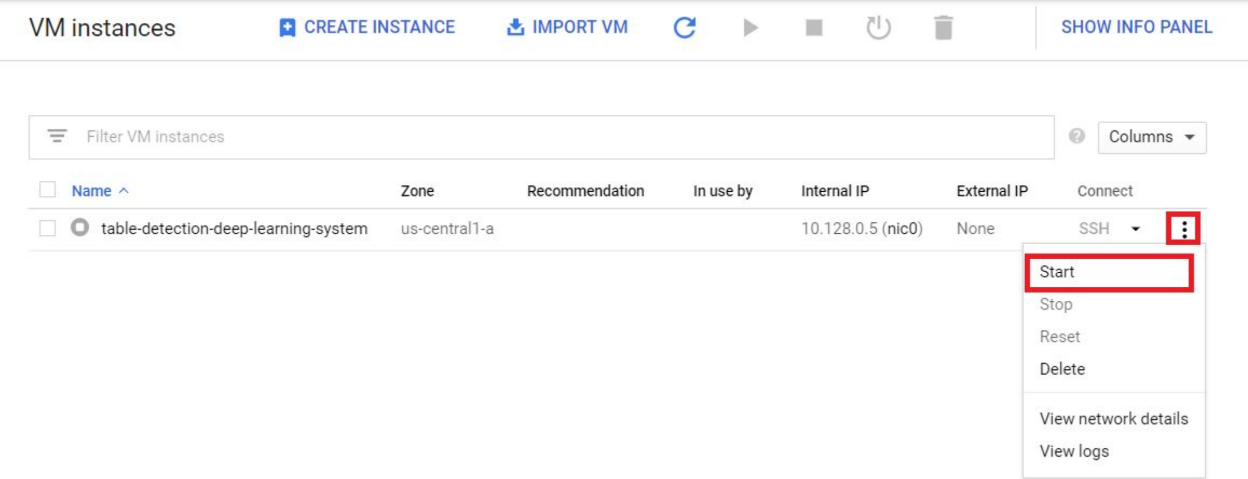
Type of key to generate:

● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

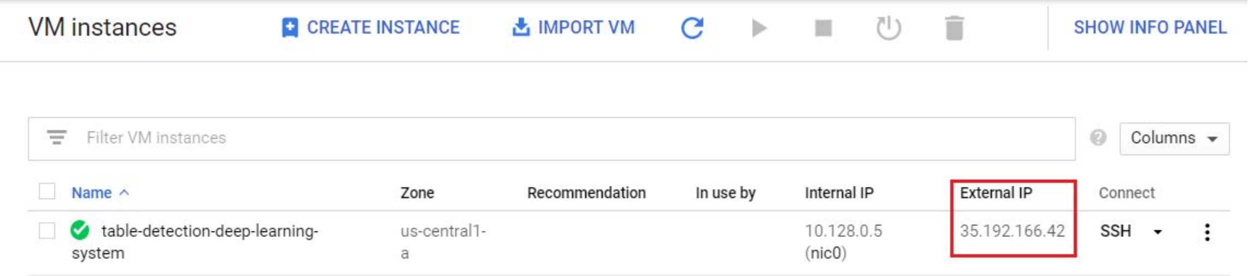
31) Go to Google Cloud console and click on “Compute Engine” -> “VM instances” from the menu

32) Start your VM instance if it's not running.



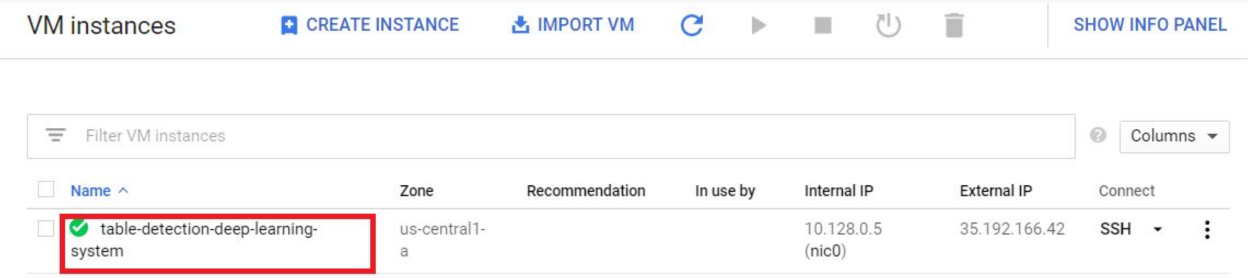
Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
table-detection-deep-learning-system	us-central1-a			10.128.0.5 (nic0)	None	SSH

33) Copy the IP Address of the VM instance from below (External IP),



Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
table-detection-deep-learning-system	us-central1-a			10.128.0.5 (nic0)	35.192.166.42	SSH

34) Click on the VM instance name,



Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
table-detection-deep-learning-system	us-central1-a			10.128.0.5 (nic0)	35.192.166.42	SSH

35) Click on “EDIT” link,

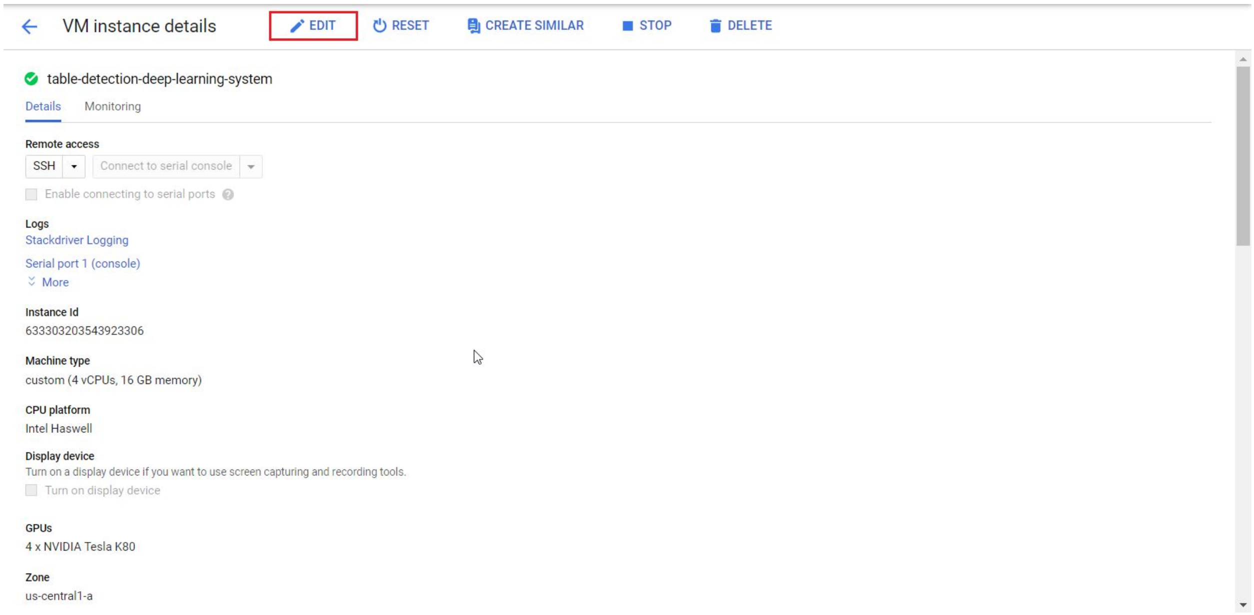


table-detection-deep-learning-system

Remote access
SSH
Connect to serial console
□ Enable connecting to serial ports

Logs
Stackdriver Logging
Serial port 1 (console)
More

Instance Id
633303203543923306

Machine type
custom (4 vCPUs, 16 GB memory)

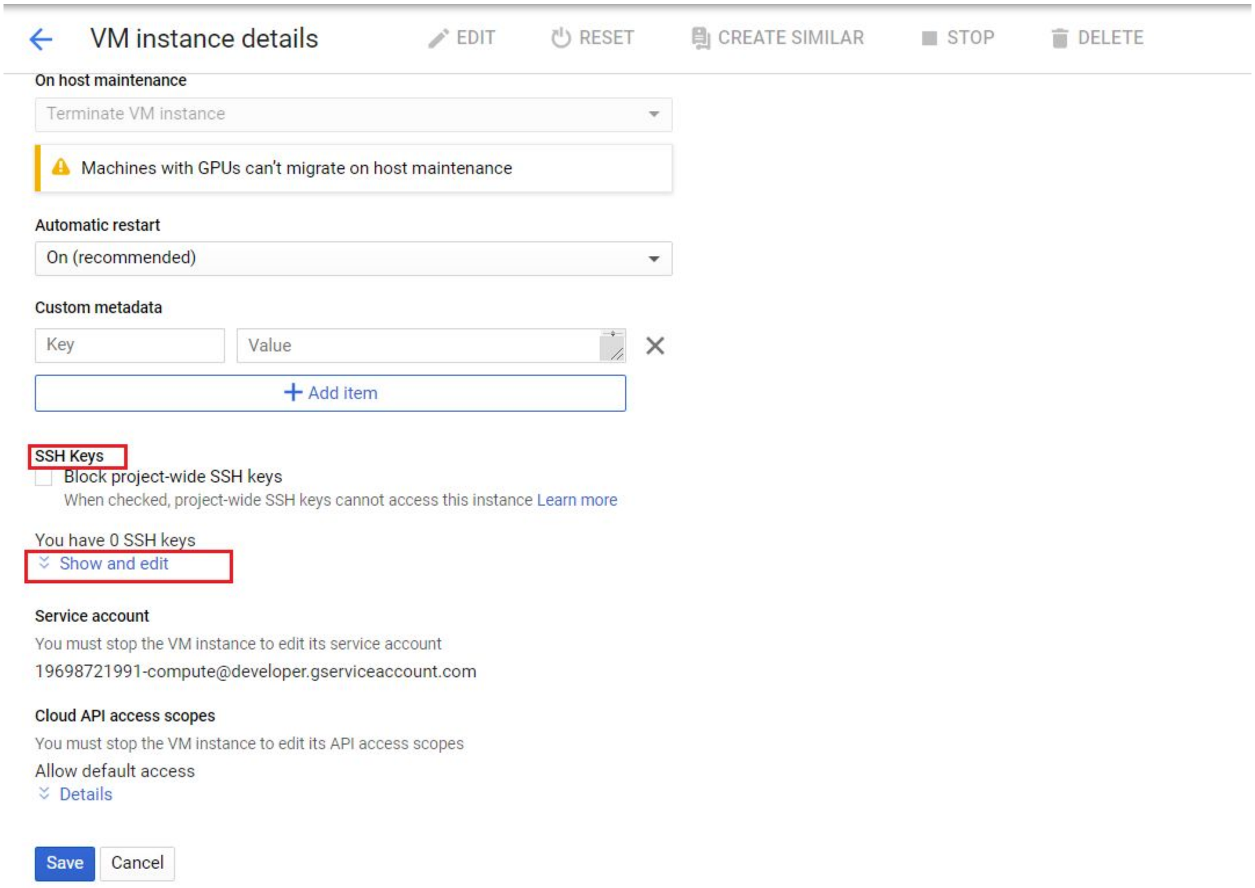
CPU platform
Intel Haswell

Display device
Turn on a display device if you want to use screen capturing and recording tools.
□ Turn on display device

GPUs
4 x NVIDIA Tesla K80

Zone
us-central1-a

36) Under the "SSH Keys” section click on the “Show and edit” link,



On host maintenance
Terminate VM instance

Machines with GPUs can't migrate on host maintenance

Automatic restart
On (recommended)

Custom metadata

SSH Keys
□ Block project-wide SSH keys
When checked, project-wide SSH keys cannot access this instance Learn more

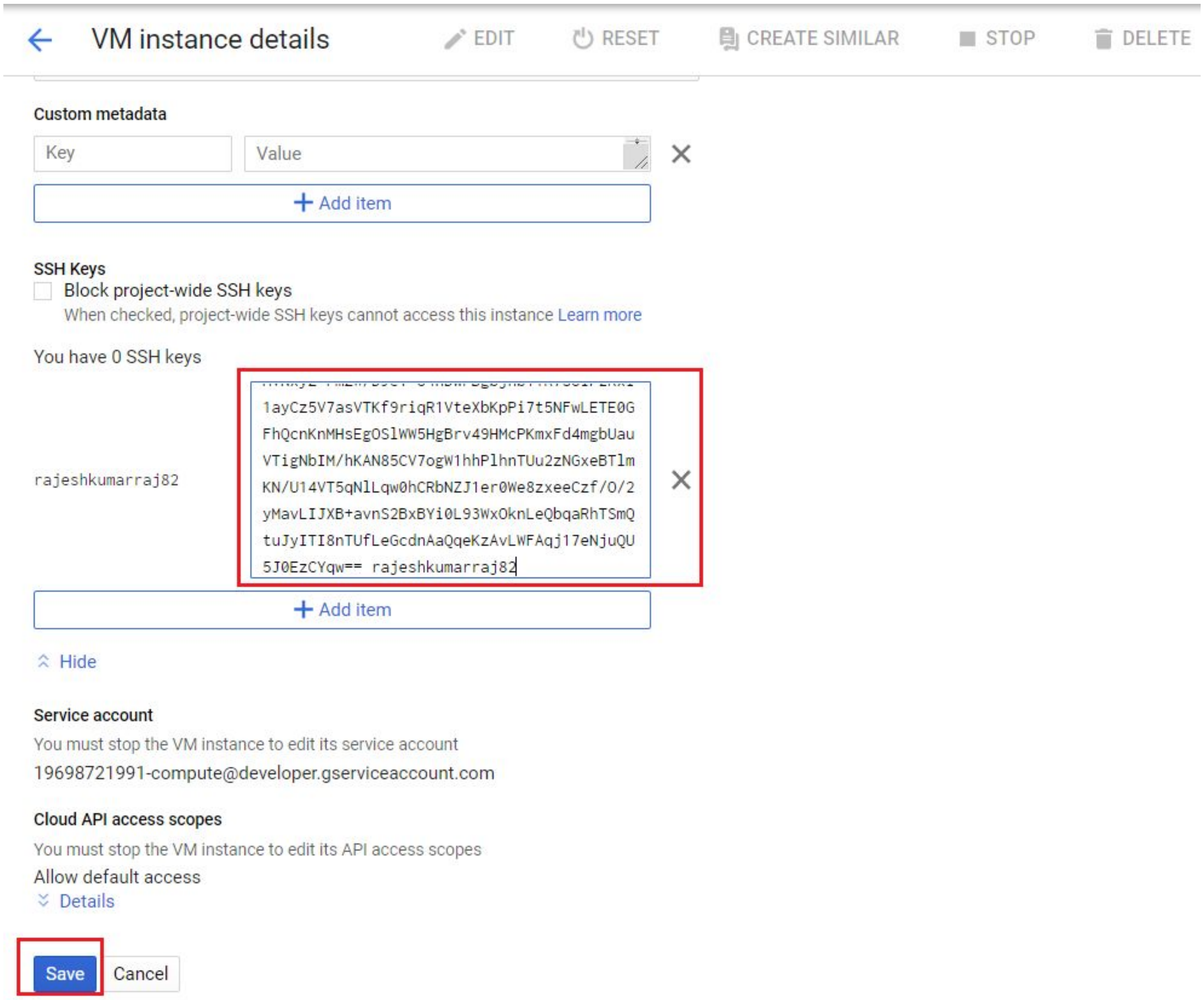
You have 0 SSH keys
Show and edit

Service account
You must stop the VM instance to edit its service account
19698721991-compute@developer.gserviceaccount.com

Cloud API access scopes
You must stop the VM instance to edit its API access scopes
Allow default access
Details

37) Paste the public key we copied above as shown below,

38) Click on the "Save" button



Custom metadata

Key | Value

SSH Keys

□ Block project-wide SSH keys

When checked, project-wide SSH keys cannot access this instance Learn more

You have 0 SSH keys

rajeshkumarraj82

```
1ayCz5V7asVTKf9riqR1VteXbKpPi7t5NFwLETE0G
FhQcnKnMHsEgOSlWW5HgBrv49HMcPKmxFd4mgbUau
VTigNbIM/hKAN85CV7ogW1hhPlhnTUu2zNGxeBTlm
KN/U14VT5qNlLqw0hCRbNZJ1er0We8zxeeCzf/O/2
yMavLIJXB+avnS2BxBYi0L93WxOknLeQbqaRhTSmQ
tuJyITI8nTUfLeGcdnAaQqeKzAvLWFAqj17eNjuQU
5J0EzCYqw== rajeshkumarraj82
```

Service account

You must stop the VM instance to edit its service account

19698721991-compute@developer.gserviceaccount.com

Cloud API access scopes

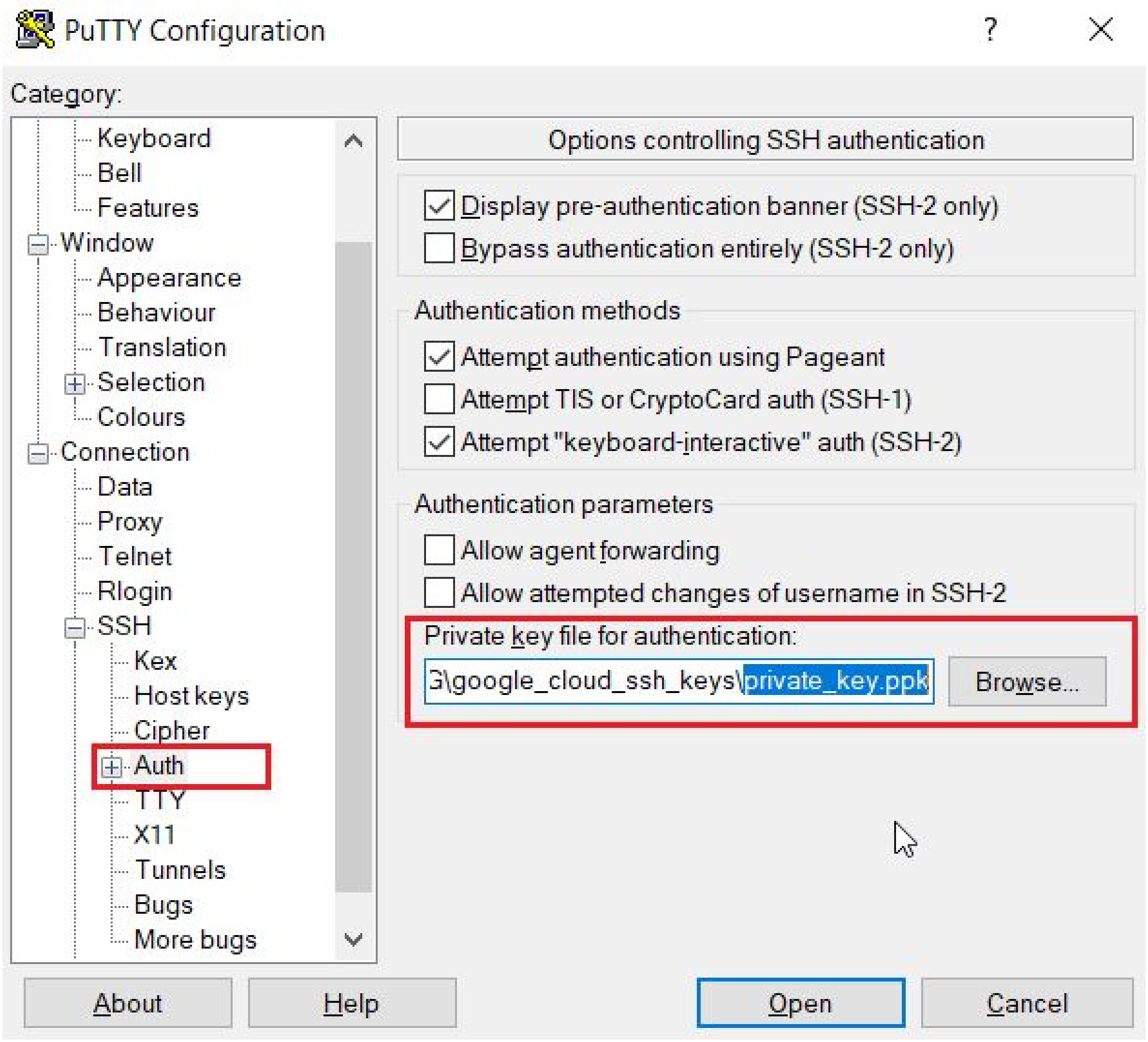
You must stop the VM instance to edit its API access scopes

Allow default access

39) Execute “putty.exe”

40) Click on the "Connection" -> “SSH” -> “Auth” option.

41) Select the “private_key.ppk” saved from above.



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)

☐ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

☐ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☐ Allow agent forwarding

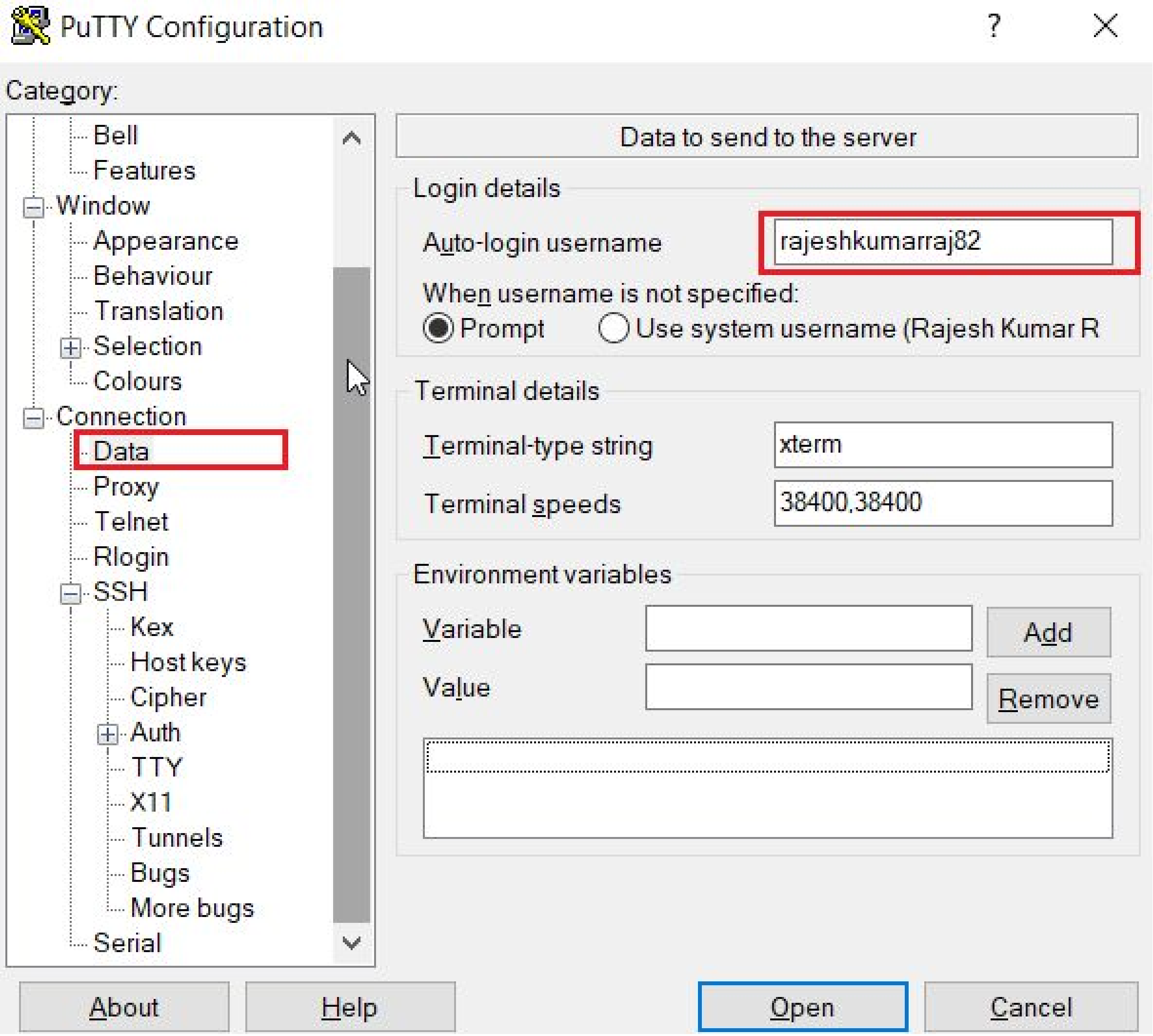
☐ Allow attempted changes of username in SSH-2

Private key file for authentication:

3\google_cloud_ssh_keys\private_key.ppk Browse...

42) Click on “Connection” -> “Data” option.

43) Enter the username specified above under “Auto-login username”



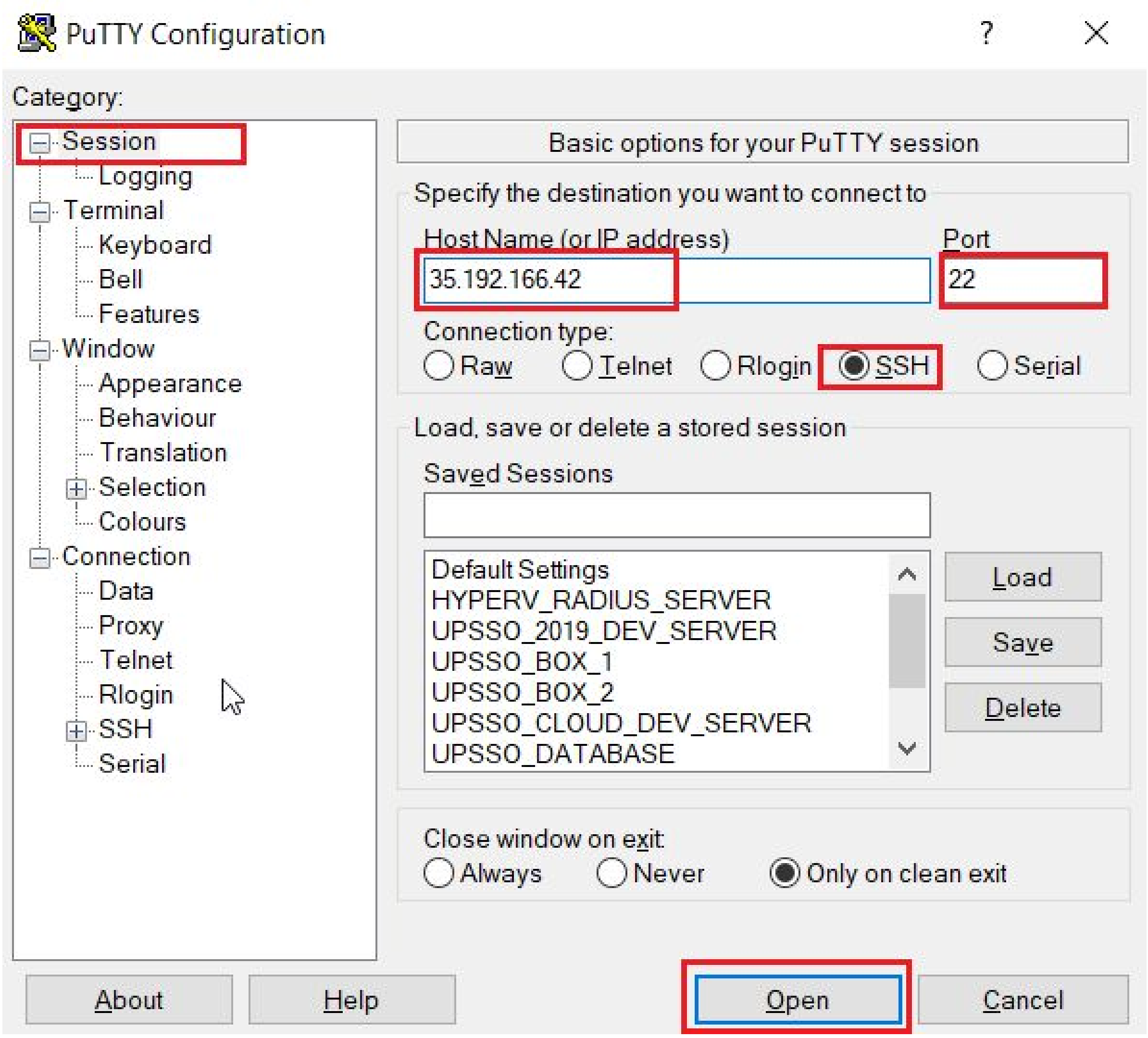
Auto-login username: rajeshkumarraj82

When username is not specified:
● Prompt ○ Use system username (Rajesh Kumar R

Terminal-type string: xterm

Terminal speeds: 38400,38400

44) Goto “Session” option
45) Enter the IP Address of the VM instance under “Host Name”
46) Enter Port as 22
47) Select “Connection type” as “SSH”
48) Click on “Open” button



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 35.192.166.42

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

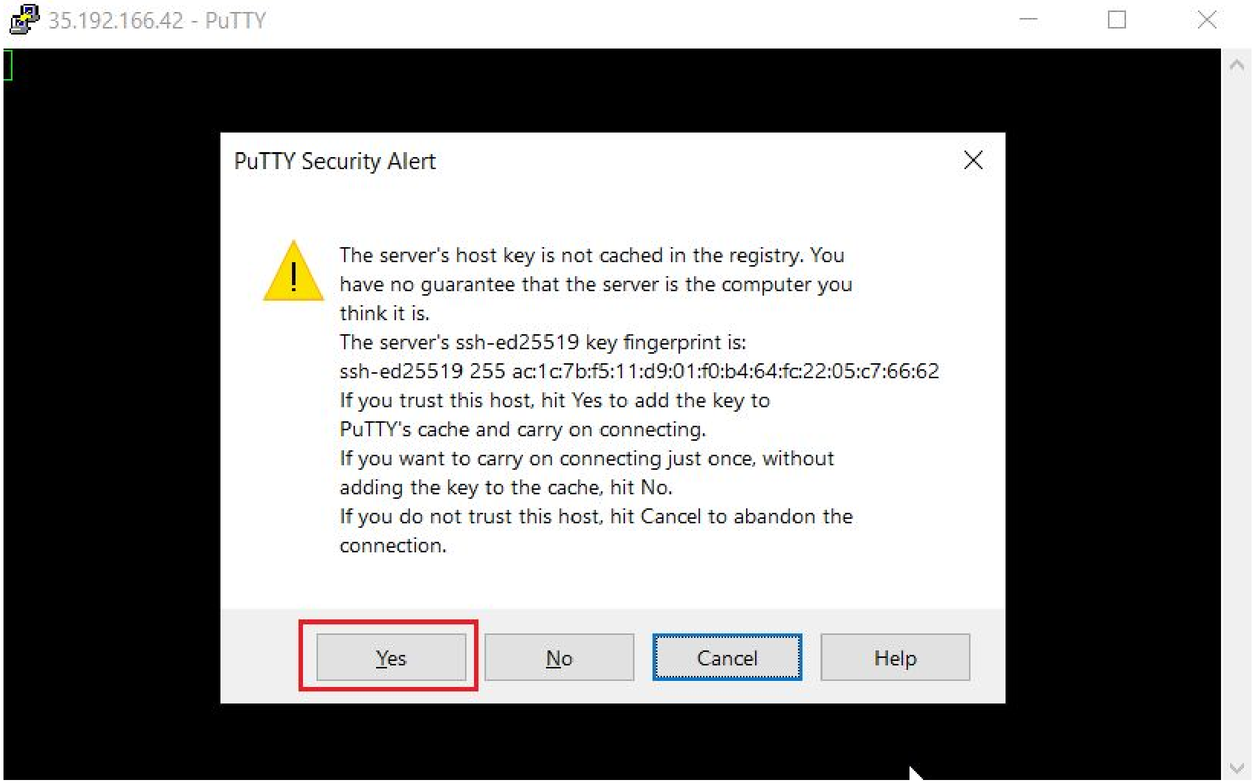
Load, save or delete a stored session

Saved Sessions

Default Settings
HYPERV_RADIUS_SERVER
UPSSO_2019_DEV_SERVER
UPSSO_BOX_1
UPSSO_BOX_2
UPSSO_CLOUD_DEV_SERVER
UPSSO_DATABASE

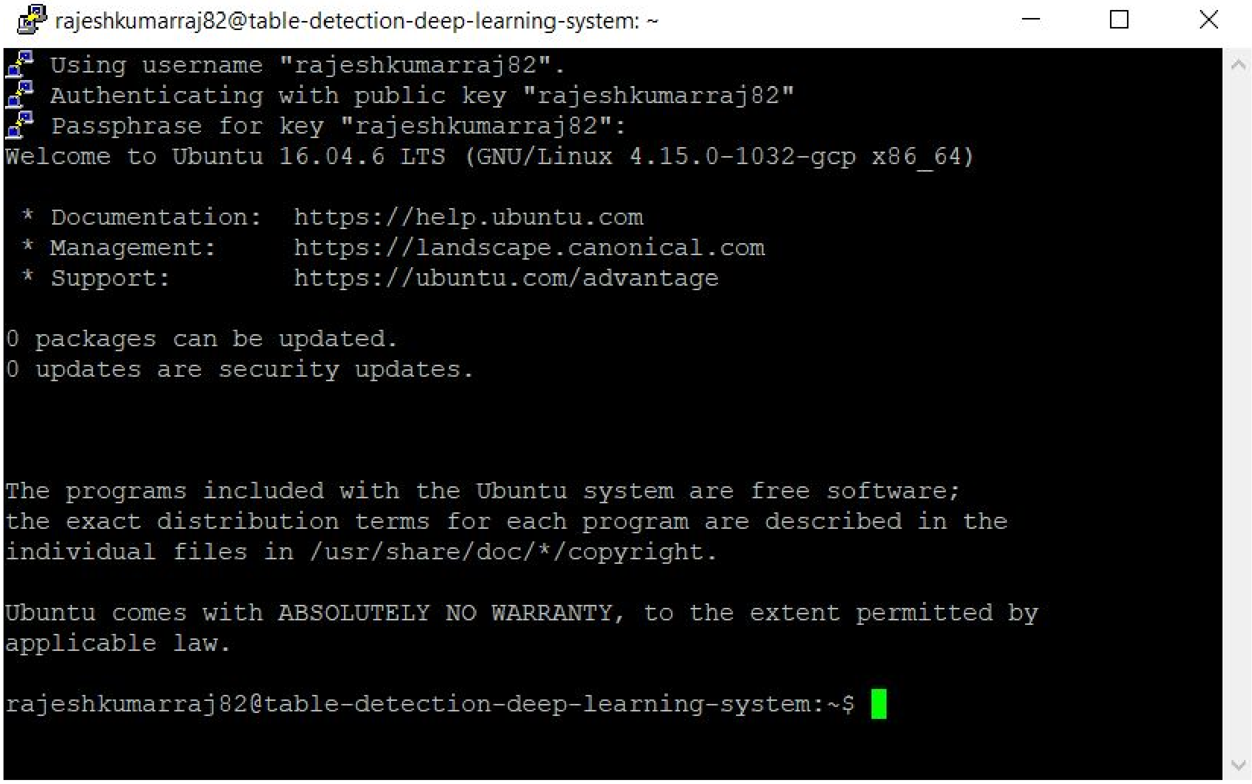
Close window on exit: ○ Always ○ Never ● Only on clean exit

49) The warning below will be displayed. Click on “Yes” button.



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 ac:1c:7b:f5:11:d9:01:f0:b4:64:fc:22:05:c7:66:62
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

50) Enter the passphrase specified above, you will be logged into the system,



```
Using username "rajeshkumarraj82".
Authenticating with public key "rajeshkumarraj82"
Passphrase for key "rajeshkumarraj82":
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1032-gcp x86_64)

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

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



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

rajeshkumarraj82@table-detection-deep-learning-system:~$
```

51) Switch to root user by typing the following command,

```
sudo su -
```

Note : Make sure you always switch to root user when executing commands in this article.

INSTALLING NVIDIA CUDA

Login into the Ubuntu Instance using putty and switch to root user,

```
sudo su -
```

52) Install Nvidia drivers by executing the following commands,

```
add-apt-repository ppa:graphics-drivers/ppa
apt-get update
apt-get install nvidia-396 nvidia-modprobe
```

53) Verify that everything is working by running,

```
nvidia-smi
```

54) Install CUDA 9.0 by executing the following commands,

```
cd /tmp

wget https://developer.nvidia.com/compute/cuda/9.0/Prod/local_installers/cuda_9.0.176_384.81_linux-run

chmod +x cuda_9.0.176_384.81_linux-run

./cuda_9.0.176_384.81_linux-run --extract=/tmp

rm NVIDIA-Linux-x86_64-384.81.run

./cuda-linux.9.0.176-22781540.run
```

This command will be prompted the following: Respond as mentioned below,

Do you accept the previously read EULA?
accept

Enter install path
Press Enter

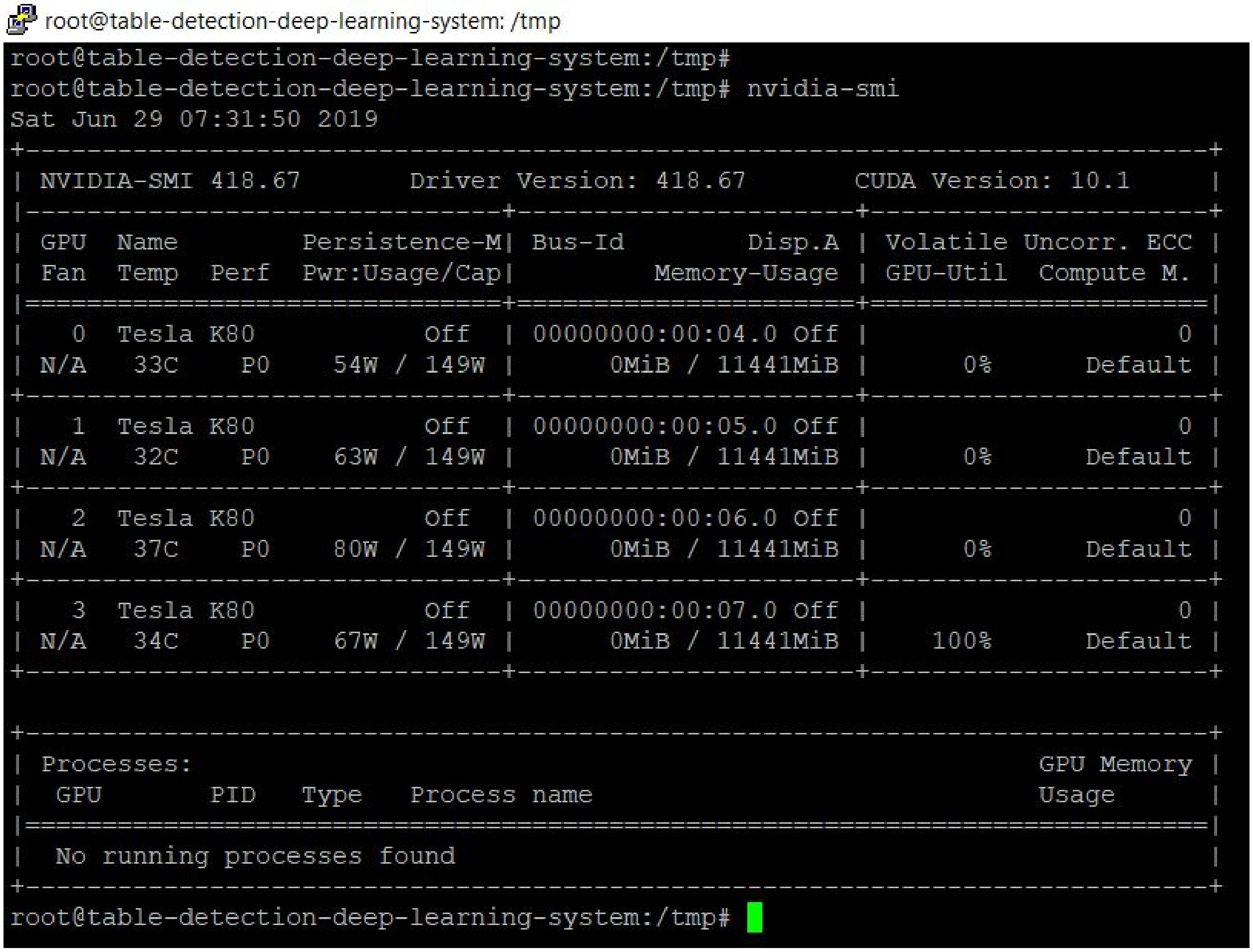
Would you like to create a symbolic link /usr/local/cuda pointing to /usr/local/cuda-9.0?
y

55) Execute the following commands to set the environment variables,

```
echo 'export CUDA_HOME=/usr/local/cuda' >> ~/.bashrc
echo 'export PATH=$PATH:$CUDA_HOME/bin' >> ~/.bashrc
echo 'export LD_LIBRARY_PATH=$CUDA_HOME/lib64' >> ~/.bashrc
source ~/.bashrc
```

56) Execute the following command to make sure the setup is successful. The below command will display the GPU’s available,

```
nvidia-smi
```



```
root@table-detection-deep-learning-system:/tmp#
root@table-detection-deep-learning-system:/tmp# nvidia-smi
Sat Jun 29 07:31:50 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.67       Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla K80           Off  | 00000000:00:04.0 Off |                    0 |
| N/A   33C    P0    54W / 149W |      0MiB / 11441MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla K80           Off  | 00000000:00:05.0 Off |                    0 |
| N/A   32C    P0    63W / 149W |      0MiB / 11441MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla K80           Off  | 00000000:00:06.0 Off |                    0 |
| N/A   37C    P0    80W / 149W |      0MiB / 11441MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla K80           Off  | 00000000:00:07.0 Off |                    0 |
| N/A   34C    P0    67W / 149W |      0MiB / 11441MiB |    100%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
root@table-detection-deep-learning-system:/tmp#
```

INSTALLING cnDNN LIBRARIES

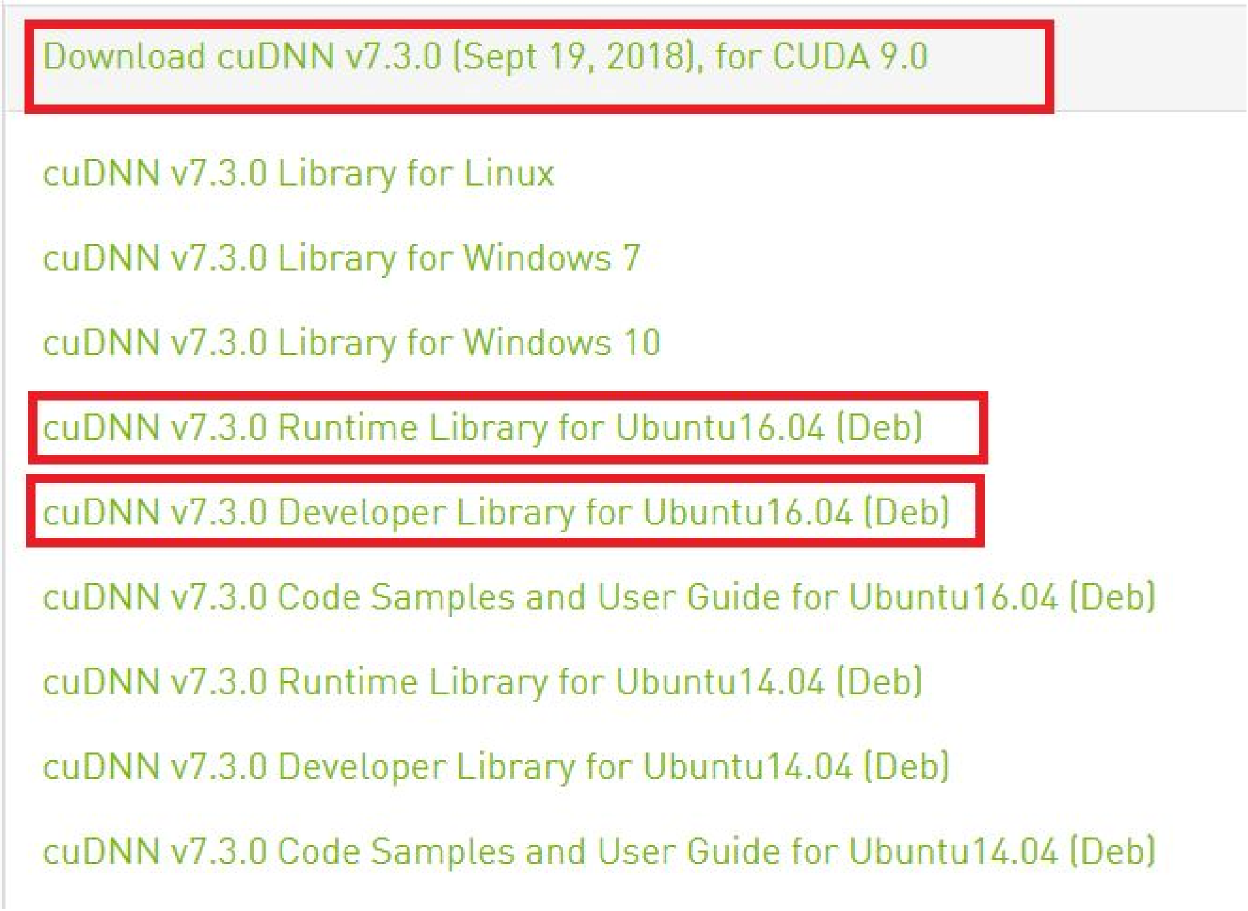
57) Register yourself at https://developer.nvidia.com/cudnn

58) Click on “Download cuDNN” button

59) Accept the “cuDNN Software License Agreement”

60) Click on “Download cuDNN v7.3.0 (Sept 19, 2018), for CUDA 9.0” link

61) Download the following:
cuDNN v7.3.0 Runtime Library for Ubuntu16.04 (Deb)
cuDNN v7.3.0 Developer Library for Ubuntu16.04 (Deb)



Download cuDNN v7.3.0 (Sept 19, 2018), for CUDA 9.0

cuDNN v7.3.0 Library for Linux

cuDNN v7.3.0 Library for Windows 7

cuDNN v7.3.0 Library for Windows 10

cuDNN v7.3.0 Runtime Library for Ubuntu16.04 (Deb)

cuDNN v7.3.0 Developer Library for Ubuntu16.04 (Deb)

cuDNN v7.3.0 Code Samples and User Guide for Ubuntu16.04 (Deb)

cuDNN v7.3.0 Runtime Library for Ubuntu14.04 (Deb)

cuDNN v7.3.0 Developer Library for Ubuntu14.04 (Deb)

cuDNN v7.3.0 Code Samples and User Guide for Ubuntu14.04 (Deb)

62) The following files will be downloaded.

libcudnn7_7.3.0.29-1+cuda9.0_amd64.deb
libcudnn7-dev_7.3.0.29-1+cuda9.0_amd64.deb

63) To add cuDNN libraries transfer the above 2 files to the /tmp directory of Ubuntu server (Using some FTP software like FileZilla or WinSCP)

64) Execute the following commands to install the cuDNN libraries,

```
dpkg -i libcudnn7_7.3.0.29-1+cuda9.0_amd64.deb
dpkg -i libcudnn7-dev_7.3.0.29-1+cuda9.0_amd64.deb
```

65) Set environment variables by executing the commands,

```
echo 'export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/cuda-9.0/lib64' >> ~/.bashrc

echo 'export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/extras/CUPTI/lib64' >> ~/.bashrc

source ~/.bashrc
```

UPGRADING PYTHON FROM VERSION 2.7 TO 3.5

66) Execute the following commands to upgrade python,

```
        add-apt-repository ppa:jonathonf/python-3.6

        apt update

        apt install python3.6

        rm /usr/bin/python

        ln -s /usr/bin/python3 /usr/bin/python

        python -V
```

INSTALLING THE REQUIRED PYTHON PACKAGES

67) Execute the following commands to install the required python packages,

```
apt install unzip

apt install python3-pip

pip3 install pillow

pip3 install pandas

pip3 install opencv-python
```

```
pip3 install tensorflow-gpu==1.8.0

pip3 install luminoth
```

SETTING UP DEVELOPMENT ENVIRONMENT & GETTING DATA

I have shared the training data and code publicly so it can be directly downloaded inside the VM instance.

68) Execute the following commands,

```
cd /usr/local

wget https://github.com/rajeshkumarraj82/table-detection-from-images-using-deep-learning/archive/master.zip

unzip master.zip
```

PREPROCESSING THE IMAGES

69) Create the following directories,

```
mkdir /usr/local/table-detection-from-images-using-deep-learning-master/data/train

mkdir /usr/local/table-detection-from-images-using-deep-learning-master/data/val
```

70) Execute the following command to preprocess the images,

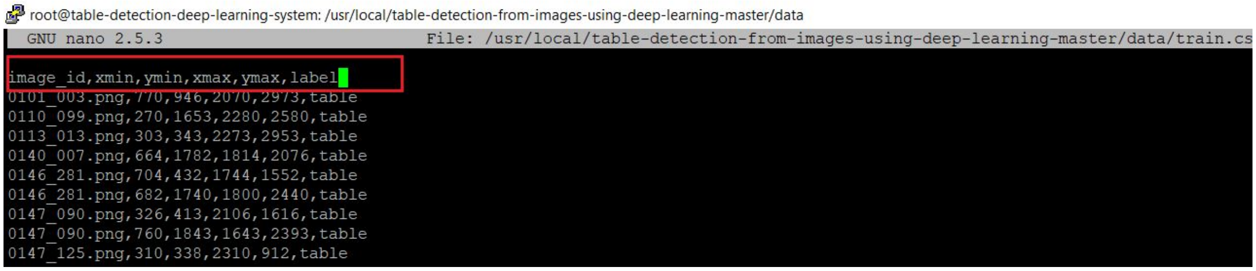
```
cd /usr/local/table-detection-from-images-using-deep-learning-master/

python preprocess.py
```

GENERATING TENSORFLOW DATA

71) Open the "/usr/local/table-detection-from-images-using-deep-learning-master/data/train.csv" file and add the following header to the first line.

image_id,xmin,ymin,xmax,ymax,label



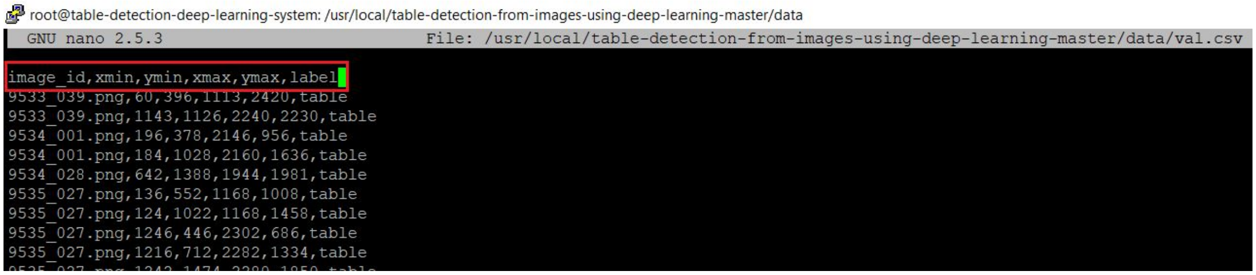
```
root@table-detection-deep-learning-system: /usr/local/table-detection-from-images-using-deep-learning-master/data
  GNU nano 2.5.3                            File: /usr/local/table-detection-from-images-using-deep-learning-master/data/train.cs

image_id,xmin,ymin,xmax,ymax,label
0101_003.png,770,946,2070,2973,table
0110_099.png,270,1653,2280,2580,table
0113_013.png,303,343,2273,2953,table
0140_007.png,664,1782,1814,2076,table
0146_281.png,704,432,1744,1552,table
0146_281.png,682,1740,1800,2440,table
0147_090.png,326,413,2106,1616,table
0147_090.png,760,1843,1643,2393,table
0147_125.png,310,338,2310,912,table
```

72) Save and close the editor.

73) Open the "/usr/local/table-detection-from-images-using-deep-learning-master/data/val.csv" file and add the following header to the first line.

image_id,xmin,ymin,xmax,ymax,label



```
root@table-detection-deep-learning-system: /usr/local/table-detection-from-images-using-deep-learning-master/data
  GNU nano 2.5.3                            File: /usr/local/table-detection-from-images-using-deep-learning-master/data/val.csv

image_id,xmin,ymin,xmax,ymax,label
9533_039.png,60,396,1113,2420,table
9533_039.png,1143,1126,2240,2230,table
9534_001.png,196,378,2146,956,table
9534_001.png,184,1028,2160,1636,table
9534_028.png,642,1388,1944,1981,table
9535_027.png,136,552,1168,1008,table
9535_027.png,124,1022,1168,1458,table
9535_027.png,1246,446,2302,686,table
9535_027.png,1216,712,2282,1334,table
9535_027.png,1242,1474,2280,1850,table
```

74) Save and close the editor

75) Execute the following command to generate TensorFlow data,

cd /usr/local/table-detection-from-images-using-deep-learning-master/

lumi dataset transform --type csv --data-dir data/ --output-dir tfdata/ --split train --split val --only-classes=table

76) Start the training process,

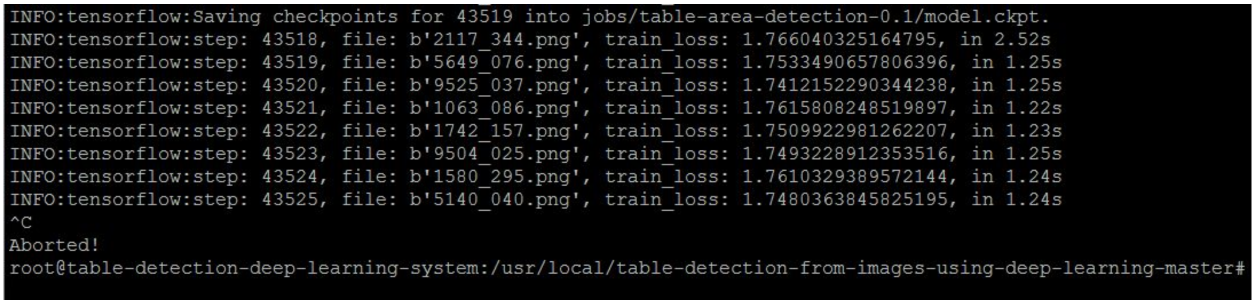
sudo su -

cd /usr/local/table-detection-from-images-using-deep-learning-master/

*chmod -R 777 **

lumi train -c config.yml

77) If the loss gets close to 1.0 you can stop training with <ctrl + c>.



```
INFO:tensorflow:Saving checkpoints for 43519 into jobs/table-area-detection-0.1/model.ckpt.
INFO:tensorflow:step: 43518, file: b'2117_344.png', train_loss: 1.766040325164795, in 2.52s
INFO:tensorflow:step: 43519, file: b'5649_076.png', train_loss: 1.7533490657806396, in 1.25s
INFO:tensorflow:step: 43520, file: b'9525_037.png', train_loss: 1.7412152290344238, in 1.25s
INFO:tensorflow:step: 43521, file: b'1063_086.png', train_loss: 1.7615808248519897, in 1.22s
INFO:tensorflow:step: 43522, file: b'1742_157.png', train_loss: 1.7509922981262207, in 1.23s
INFO:tensorflow:step: 43523, file: b'9504_025.png', train_loss: 1.7493228912353516, in 1.25s
INFO:tensorflow:step: 43524, file: b'1580_295.png', train_loss: 1.7610329389572144, in 1.24s
INFO:tensorflow:step: 43525, file: b'5140_040.png', train_loss: 1.7480363845825195, in 1.24s
^C
Aborted!
root@table-detection-deep-learning-system:/usr/local/table-detection-from-images-using-deep-learning-master#
```

CREATE CHECKPOINT

78) Execute the below command to create check point,

```
lumi checkpoint create config.yml
```

The response will be like,

```
Creating checkpoint for given configuration...
Checkpoint c2df81db49e0 created successfully.
```

79) Take a note of the Checkpoint number above.

TESTING PREDICTING TABLE LOCATION IN A SAMPLE IMAGE

80) Execute the below command to predict the location of the table in a random image (Use the Checkpoint number noted in the above section)

```
lumi predict --checkpoint c2df81db49e0 data/val/9541_023.png
```

81) The response will be having the coordinates of the tables identified,

```
Predicting data/val/9541_023.png... done.
{"file": "data/val/9541_023.png", "objects": [{"label": "table", "prob": 1.0, "bbox": [121, 613, 2398, 2148]}]}
```

PREDICTING TABLE LOCATION USING A PYTHON PROGRAM

82) Here’s the python code that will call Luminoth API to predict the table location from an image, (Please note you need to specify the Checkpoint number & image file in this code)

File Name : predict_table_location_from_image.py

```
from luminoth.tools.checkpoint import get_checkpoint_config
from luminoth.utils.predicting import PredictorNetwork
from PIL import Image as pilimage

# This program  will predict the location of the tables in an image
# It outputs the coordinates of the tables. Using these coordinates we can cut the table portion of the image and use it for further processing
```

```
input_file = '/usr/local/table-detection-from-images-using-deep-learning-master/test_image_with_table.png'
# Specify the luminoth checkpoint here
checkpoint = 'c2df81db49e0'

config = get_checkpoint_config(checkpoint)
network = PredictorNetwork(config)
image = pilimage.open(input_file).convert('RGB')
objects = network.predict_image(image)

print("NO OF TABLES IDENTIFIED BY LUMINOTH = " + str(len(objects)))
print('-' * 100)

table_counter = 1

for i in range(len(objects)):
    table_idctionary = objects[i]
    coordinate_list = table_idctionary["bbox"]
    xminn = coordinate_list[0]
    yminn = coordinate_list[1]
    xmaxx = coordinate_list[2]
    ymaxx = coordinate_list[3]
    print('TABLE ' + str(table_counter) + ':')
    print('-' * 100)
    print("xminn = " + str(xminn))
    print("yminn = " + str(yminn))
    print("xmaxx = " + str(xmaxx))
    print("ymaxx = " + str(ymaxx))
    table_counter += 1
```

83) Execute the above python program by,

```
python predict_table_location_from_image.py
```

84) The output of the above program will be like,

```
NO OF TABLES IDENTIFIED BY LUMINOTH = 1
----------------------------------------------------------------------------------------------------
TABLE 1:
----------------------------------------------------------------------------------------------------
xminn = 254
yminn = 436
xmaxx = 1058
ymaxx = 549
```

CONCLUSION

So using the coordinates of the tables we are able to crop table portions from the images. These cropped images can be further processed with APIs like Tabula to extract table text. Cropping the table portions from images will greatly improve the accuracy of table data.