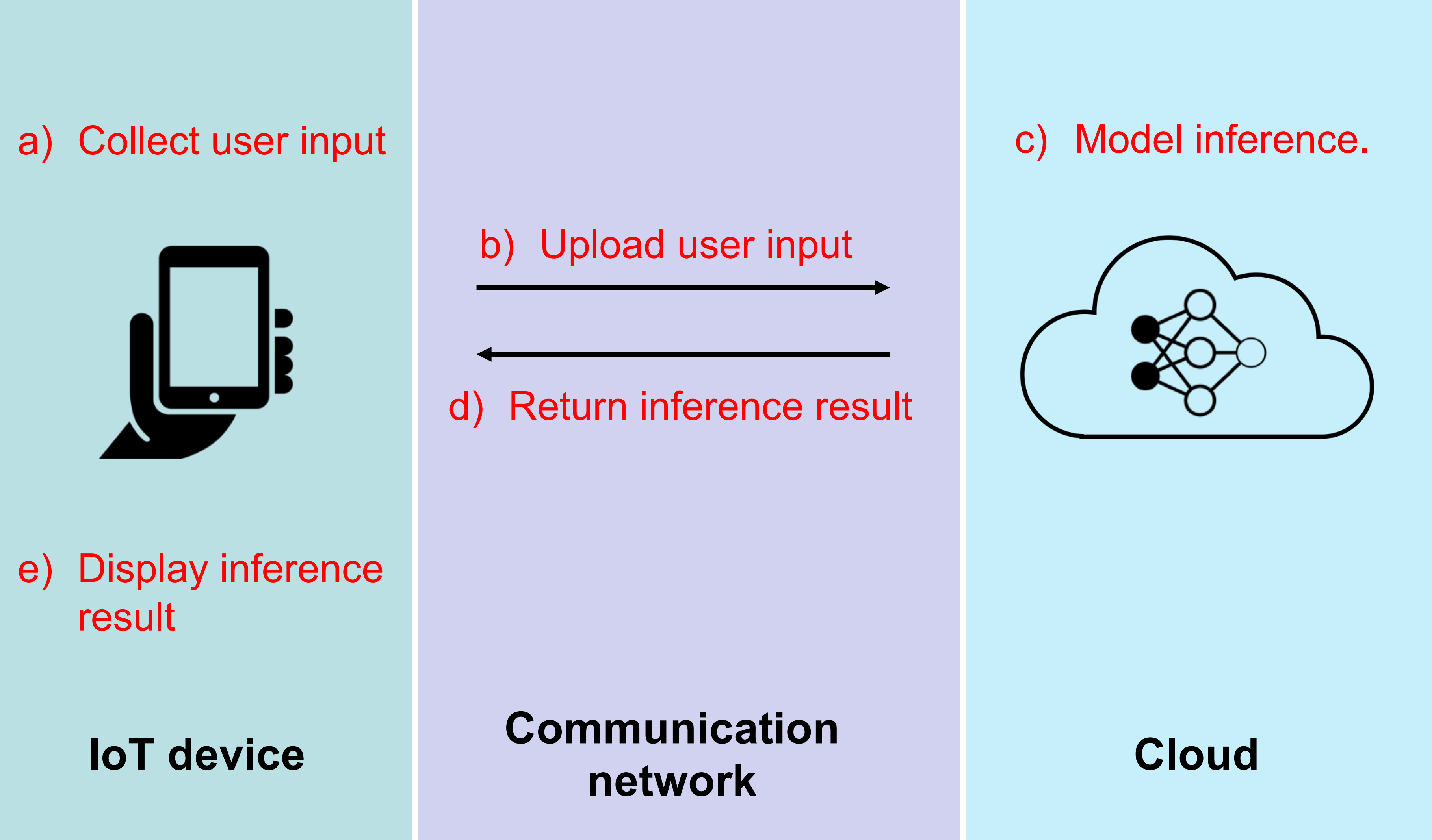
Course Project (35%)

- Offloading AI inference from IoT device to cloud
  - IoT devices are often incapable of AI inference
  - Offload computing from IoT device to server
- Details to be given in Week 3 lecture time
- Project submission
  - Deadline: May 4th 2022
  - Source code package
  - URL to a video demo with max length of 10 minutes (youtube preferred, keep the video at the URL until Oct 31st 2022)

Ultimate Goal



a) Collect user input

b) Upload user input

c) Model inference.

d) Return inference result

e) Display inference result

IoT device

Communication network

Cloud

Tips

- You may choose any applications.
  - E.g., speech recognition, image classification, etc.
- Any mobile platform for the IoT app, e.g., Android, iOS.
  - If you don't have a physical device, you may use the phone emulator.
- You may use the VM on Google Could Platform or your own PC to host the cloud service.
  - Google Cloud: 90-day, $300 Free Trial
  - **You're responsible for managing your Google Cloud account to avoid any monetary charge**
  - **In particular, close/remove the billing account after the project**
  - **If you're unsure, use your own PC to host the service**
- Any programming language for the server app
  - Python + Flask is recommended.

Creating a VM on Google Cloud Platform

- Visit the following URL and register an account
  - https://cloud.google.com/compute
- Follow the guides to create the VM of Linux / Windows
  - https://cloud.google.com/compute/docs/quickstarts

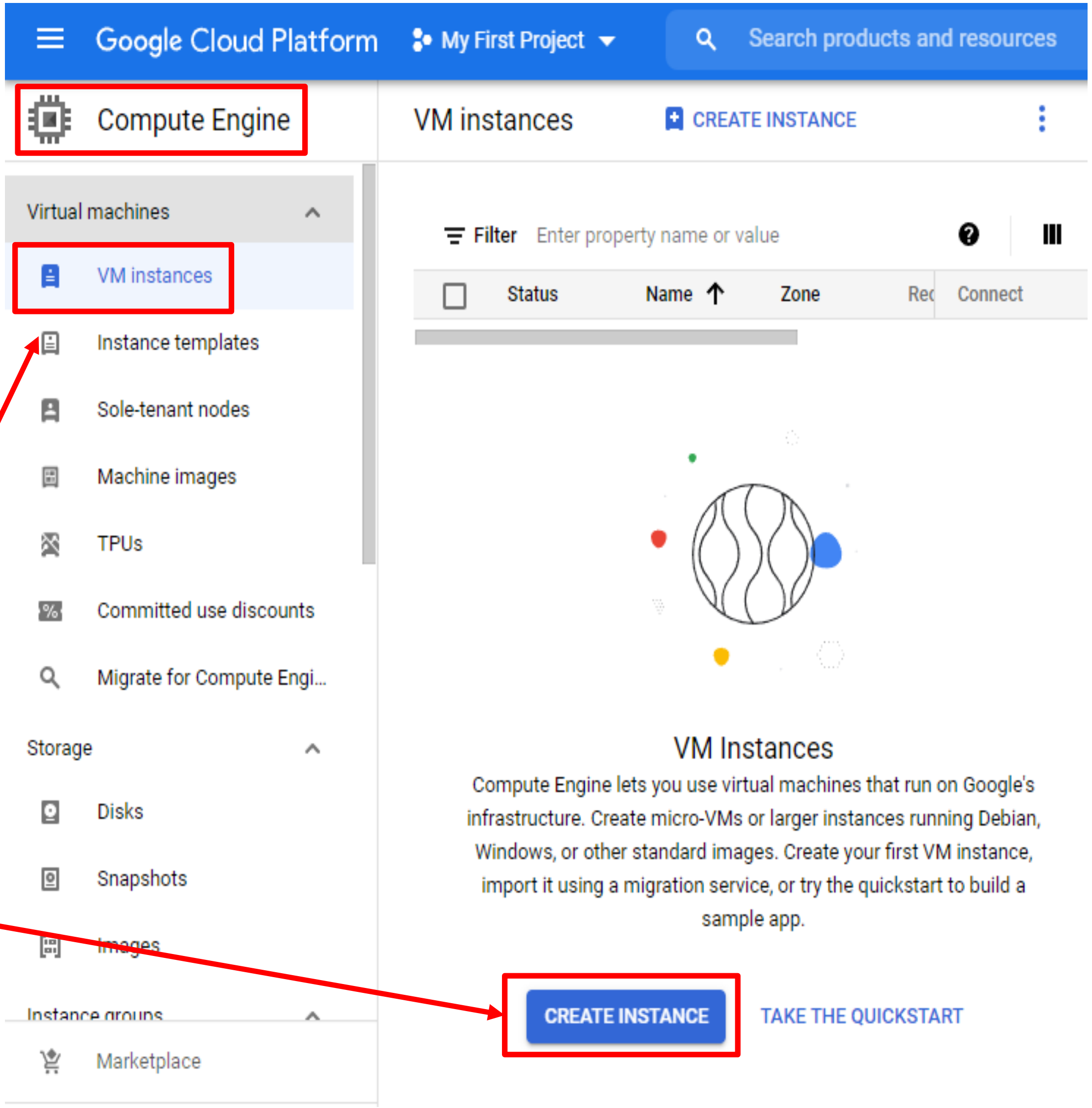
Creating a VM on Google Cloud Platform

i. Log in to home page

ii. Find “Computing Engine”

iii. Search for “VM instances”

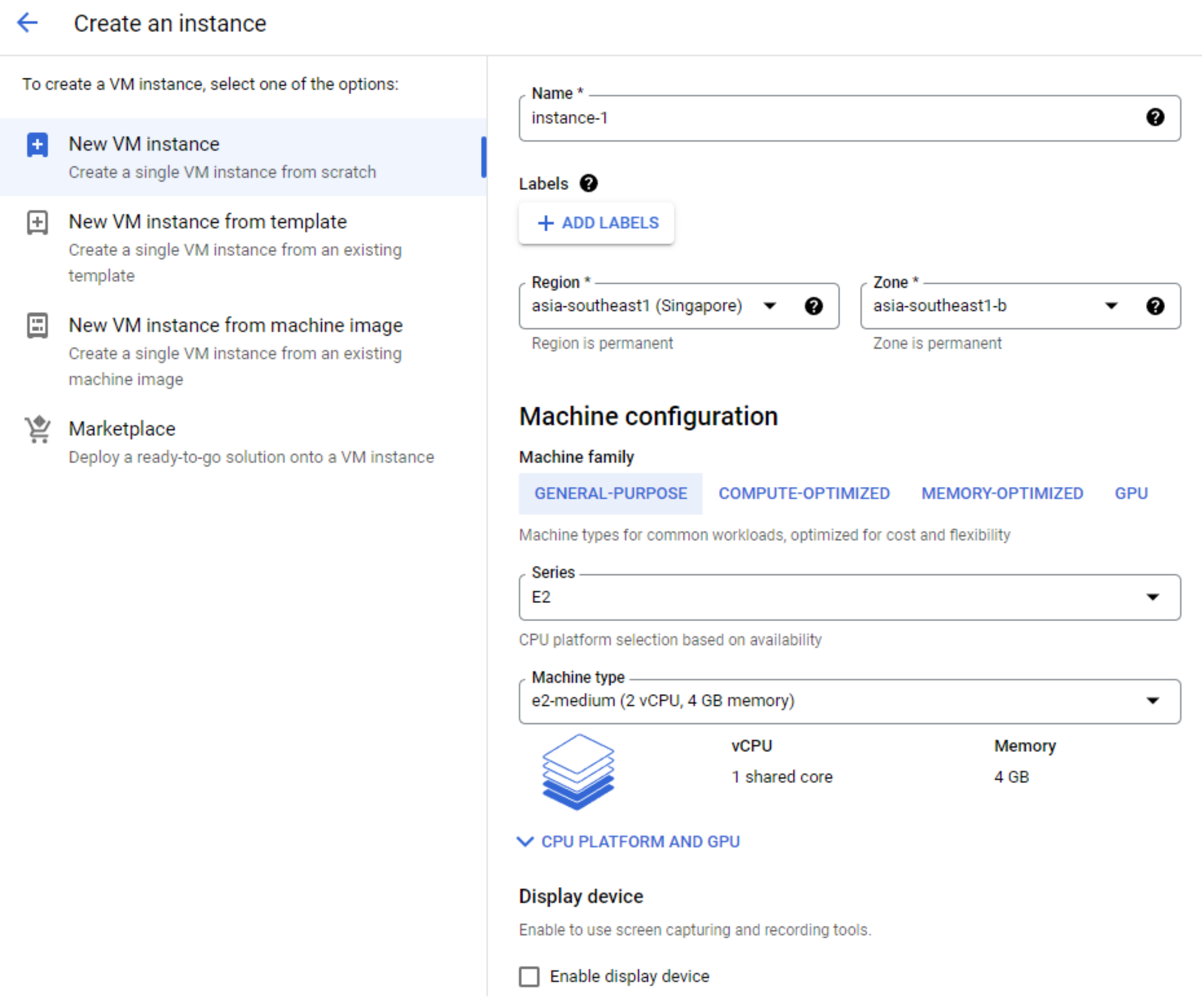
iv. Choose “Create Instance".



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.

Creating a VM on Google Cloud Platform

- Follow the steps to create the VM



Create an instance

To create a VM instance, select one of the options:

New VM instance
Create a single VM instance from scratch

New VM instance from template
Create a single VM instance from an existing template

New VM instance from machine image
Create a single VM instance from an existing machine image

Marketplace
Deploy a ready-to-go solution onto a VM instance

Name * : instance-1

Labels

+ ADD LABELS

Region * : asia-southeast1 (Singapore)
Region is permanent

Zone * : asia-southeast1-b
Zone is permanent

Machine configuration

Machine family

GENERAL-PURPOSE | COMPUTE-OPTIMIZED | MEMORY-OPTIMIZED | GPU

Machine types for common workloads, optimized for cost and flexibility

Series: E2
CPU platform selection based on availability

Machine type: e2-medium (2 vCPU, 4 GB memory)

vCPU	Memory
1 shared core	4 GB

CPU PLATFORM AND GPU

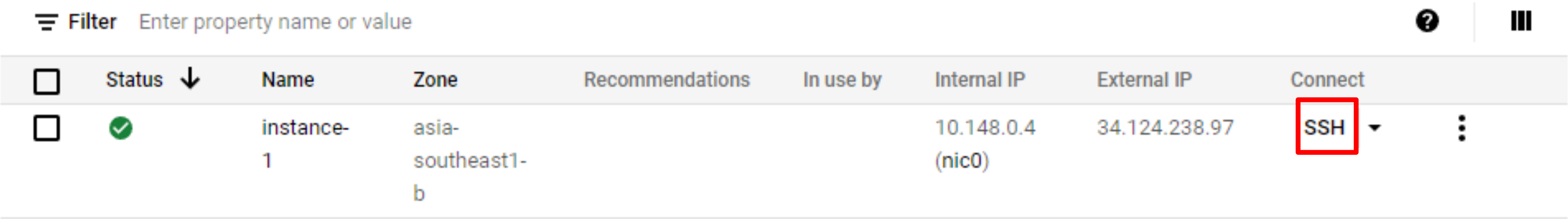
Display device

Enable to use screen capturing and recording tools.

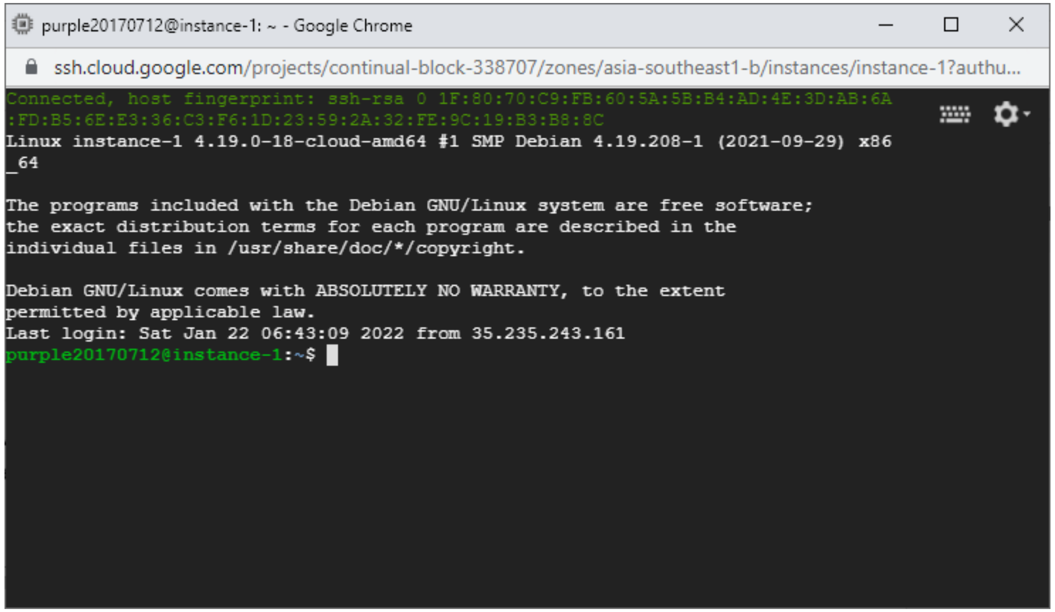
□ Enable display device

Creating a VM on Google Cloud Platform

- Check status of VM
  - Find VM's IP for your setup
  - Click "SSH" to connect the VM



□	Status ↓	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	instance-1	asia-southeast1-b			10.148.0.4 (nic0)	34.124.238.97	SSH



purple20170712@instance-1: ~ - Google Chrome
ssh.cloud.google.com/projects/continual-block-338707/zones/asia-southeast1-b/instances/instance-1?authu...

```
Connected, host fingerprint: ssh-rsa 0 1F:80:70:C9:FB:60:5A:5B:B4:AD:4E:3D:AB:6A
:FD:B5:6E:E3:36:C3:F6:1D:23:59:2A:32:FE:9C:19:B3:B8:8C
Linux instance-1 4.19.0-18-cloud-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86
_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jan 22 06:43:09 2022 from 35.235.243.161
purple20170712@instance-1:~$
```

Tasks (100 marks)

• Local inference	50 marks
– Collect user input	(15)
– Infer locally and display result	(20)
– Run on emulated/physical IoT device	(15)
• Cloud inference	30 marks
– Run inference in cloud virtual machine	(10)
– Communicate btw IoT device & cloud	(20)
• Advanced tasks	capped at 20 marks
– **If** train your own model	(10)
– **If** Support multiple concurrent users	(10)
– **If** support online model updating	(10)

Marking Criteria (1)

- Collect user input (15)
  - **If** load input from storage (8)
  - **If** collect real-time input by touch screen, microphone, or camera (15)

Marking Criteria (2)

- Infer locally and display result (20)
  - **If** run model inference on the mobile app (15)
    - If your app offloads the inference to cloud, you have the marks for this part automatically
  - **If** run heuristic algorithm (not neural network) on the mobile app (9)
    - If your app offloads the execution to cloud, you have the marks for this part automatically
  - Display the inference result in real time by screen or synthetic voice (5)

Marking Criteria (3)

- Run on emulated/physical IoT device (15)
  - **If** the program runs natively on a desktop or laptop computer (5)
  - **If** the program runs on an emulated or a physical IoT device (15)
    - iOS emulator provided by Xcode
    - Android emulator
    - Real smartphone
    - Raspberry Pi + add-on sensors (camera, microphone, etc)

Marking Criteria (4)

- Run inference in cloud virtual machine (10)
  - **If** deploy server program on a cloud virtual machine, e.g., Azure. (10)
  - **If** deploy server program on your own computer (10)

Marking Criteria (5)

- Communication btw IoT device and cloud (20)
  - Send the user input from the mobile app to the cloud for inference (10)
  - Send the inference result from the cloud to the mobile app (10)

Marking Criteria (6)

- Train your own model (10)
  - **If** train the used ML model by yourself (10)
    - Training program should be in the code package
    - Training results (e.g., accuracy) in a readme file
    - The training data can be any publicly available dataset
  - **If** use downloaded pre-trained model (8)
  - **If** use heuristic algorithm (not neural network) (6)

Marking Criteria (7)

- Support multiple concurrent users (10)
  - Demonstrate multiple IoT devices can use the cloud service simultaneously

Marking Criteria (8)

- Online model updating (10)
  - Demonstrate that the model in the cloud can be updated at run time using newly collected user data
    - User input and the corresponding label are transmitted to cloud
    - The cloud retrains the model

Grading

- Project should be independently accomplished by each student.
- Q&A and investigation may be conducted on similar topics and implementations
- 3-day grace period
  - No penalty if a valid excuse provided
  - Otherwise, a penalty of 20% reduction will be applied to the mark of the late submissions
- Submissions after the grace period
  - Zero mark