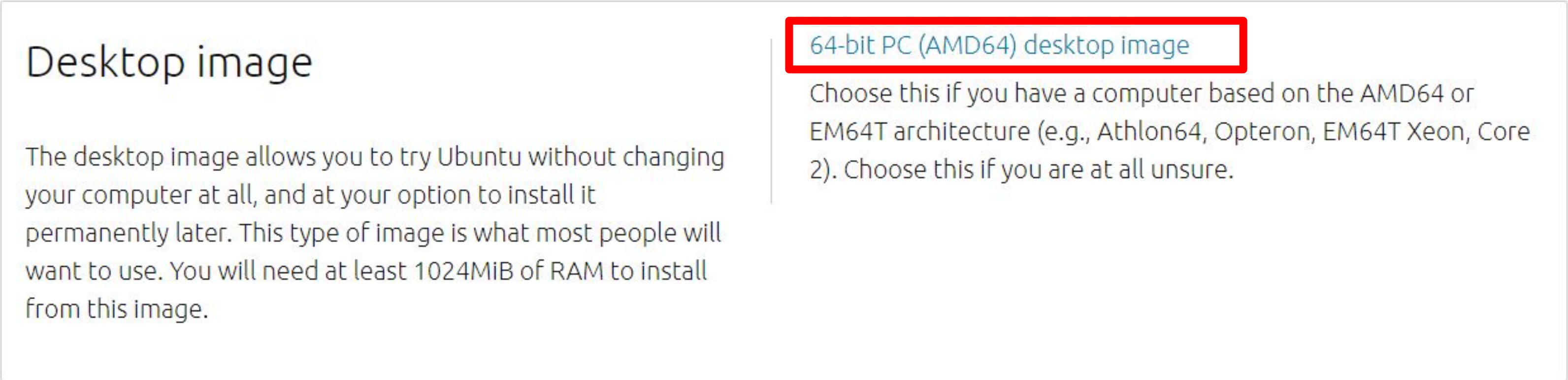
2024 Summer Course
Week 1

Presenter: Pao-Hsun, Chen

Ubuntu 20.04.6 LTS iso download

Website:https://releases.ubuntu.com/focal/



Desktop image

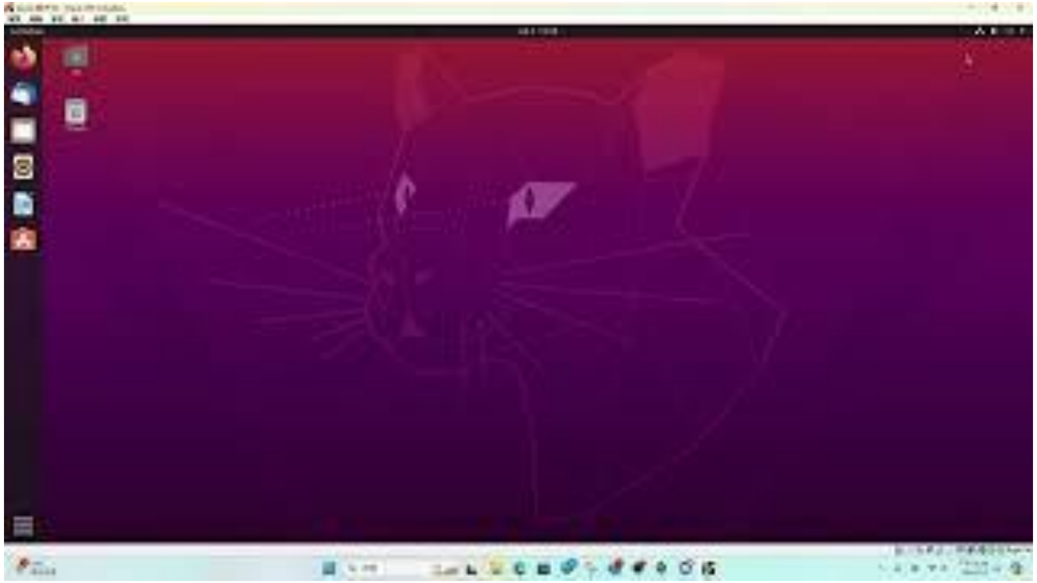
The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

Ubuntu Installation

- sudo
- sudo apt update
- sudo apt upgrade
- sudo apt install <pkg>
- cd
  - Change Direcotry
- ls
  - List
- su -
- rm -rf <dir>
  - rm -rf /



Install Ubuntu 20.04 virtual machine (youtube.com)

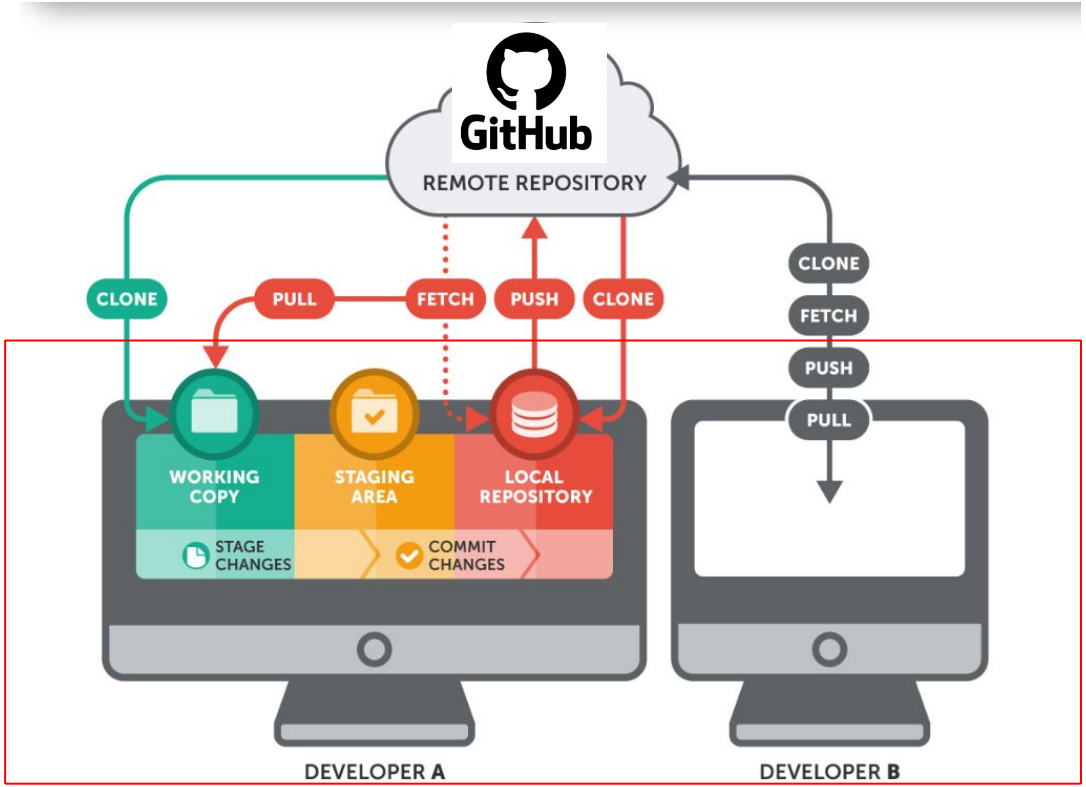
https://www.youtube.com/watch?v=VGhx7WkSUH8

Docker Introduction

- docker pull
- docker build -t <image tag> .
- docker run <image tag>
- docker image ls
- docker container ls

Git Introduction

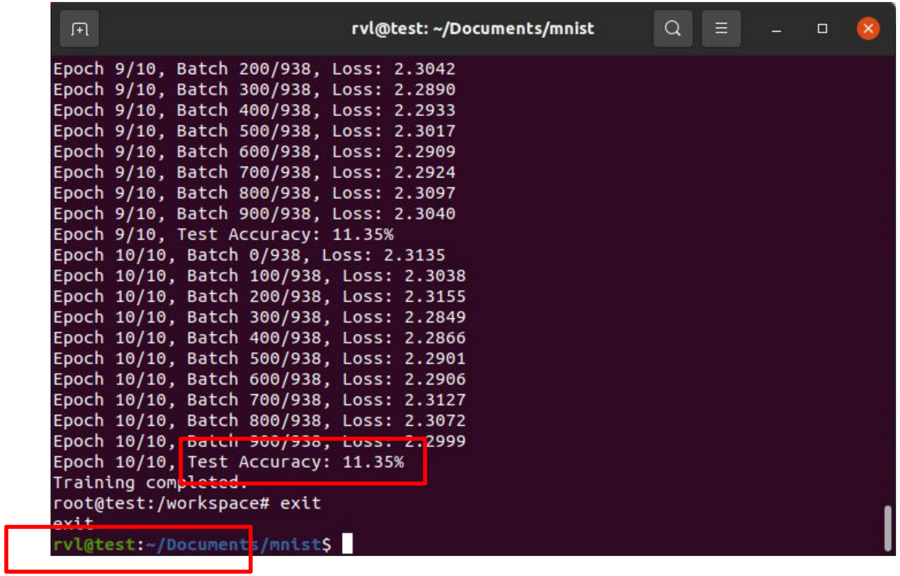
- git pull <git url>
- git commit
- git push
- git clone



Local Repository

Homework 1

Please train a model to classify mnist dataset, and you must train your model in docker container pytorch/pytorch.



```
Epoch 9/10, Batch 200/938, Loss: 2.3042
Epoch 9/10, Batch 300/938, Loss: 2.2890
Epoch 9/10, Batch 400/938, Loss: 2.2933
Epoch 9/10, Batch 500/938, Loss: 2.3017
Epoch 9/10, Batch 600/938, Loss: 2.2909
Epoch 9/10, Batch 700/938, Loss: 2.2924
Epoch 9/10, Batch 800/938, Loss: 2.3097
Epoch 9/10, Batch 900/938, Loss: 2.3040
Epoch 9/10, Test Accuracy: 11.35%
Epoch 10/10, Batch 0/938, Loss: 2.3135
Epoch 10/10, Batch 100/938, Loss: 2.3038
Epoch 10/10, Batch 200/938, Loss: 2.3155
Epoch 10/10, Batch 300/938, Loss: 2.2849
Epoch 10/10, Batch 400/938, Loss: 2.2866
Epoch 10/10, Batch 500/938, Loss: 2.2901
Epoch 10/10, Batch 600/938, Loss: 2.2906
Epoch 10/10, Batch 700/938, Loss: 2.3127
Epoch 10/10, Batch 800/938, Loss: 2.3072
Epoch 10/10, Batch 900/938, Loss: 2.2999
Epoch 10/10, Test Accuracy: 11.35%
Training completed.
root@test:/workspace# exit
exit
rvl@test:~/Documents/mnist$
```

- You must display your test accuracy and your system name in the screenshot.

Homework 2

Please solve these 2 problems with **c++ only**, and submit your code to your github repository.

- Problem 1
- Problem 2
- Problem 3

You must commit your solutions, a screenshot of your result in your terminal and a document to explain your code, analyze the time complexity and describe how to run your code then push your commit to your github repository.

About your solution file name, please follow the following patterns.

Pattern: Problem_<problem number>_sol

For example: Problem_1_sol.cpp

Problem 1.

Description:

Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value.If target is not found in the array, return [-1, -1]. You must write an algorithm with O(log n) runtime complexity.

Example 1:
Input: nums = [5,7,7,8,8,10], target = 8
Output: [3,4]

Example 2:
Input: nums = [5,7,7,8,8,10], target = 6
Output: [-1,-1]

Example 3:
Input: nums = [], target = 0
Output: [-1,-1]

Constraints:

- 0 <= nums.length <= 10^5
- -10^9 <= nums[i] <= 10^9
- nums is a non-decreasing array.
- -10^9 <= target <= 10^9

Problem 2.

Description:

Given a string s, return the longest palindromic substring in s.

Example 1:
Input: s = "babad"
Output: "bab"
Explanation: "aba" is also a valid answer.

Example 2:
Input: s = "cbbd"
Output: "bb"

Constraints:

- 1 <= s.length <= 1000
- s consist of only digits and English letters.

Problem 3.

Description:

Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays.

The overall run time complexity should be O(log (m+n)).

Example 1:
Input: nums1 = [1,3], nums2 = [2]
Output: 2.00000
Explanation: merged array = [1,2,3] and median is 2.

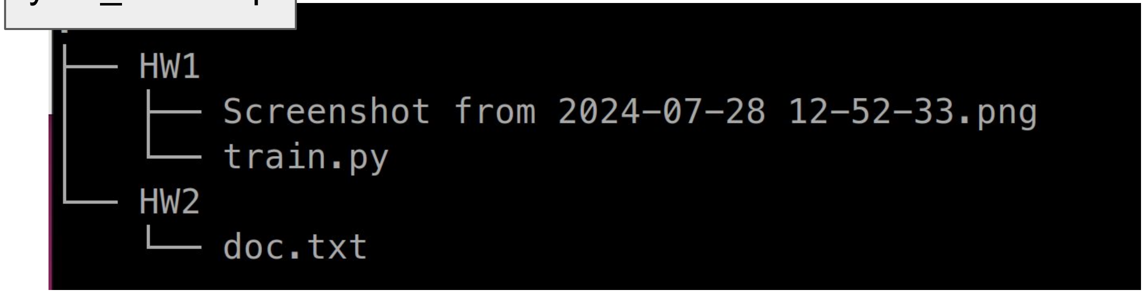
Example 2:
Input: nums1 = [1,2], nums2 = [3,4]
Output: 2.50000
Explanation: merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5.

Constraints:

- nums1.length == m
- nums2.length == n
- 0 <= m <= 1000
- 0 <= n <= 1000
- 1 <= m + n <= 2000
- -10^6 <= nums1[i], nums2[i] <= 10^6

Deadline: 8/8 23:59

your_name.zip



```
├── HW1
│   ├── Screenshot from 2024-07-28 12-52-33.png
│   └── train.py
└── HW2
    └── doc.txt
```