1. AWS connection:

Use pem file and ipv4 dns for connection

use security-group-…. Add ip address when using new devices

1. Mobaxterm ssh link:

ubuntu@ip-10-0-0-9:~$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/ubuntu/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/ubuntu/.ssh/id\_rsa

Your public key has been saved in /home/ubuntu/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:CurVynb1qChdXYMVlDRJ3EU4ZKAK14zBZTszKYjUvQI ubuntu@ip-10-0-0-9

The key's randomart image is:

+---[RSA 3072]----+

| .. o..o=BBo+o |

| .E...o\* +=o+ |

| ..o +.Xo . |

| .o.o.+o |

| ....S. . |

| . o..o |

| ....o. o |

| ..ooo. . . |

| .o+... |

+----[SHA256]-----+

ubuntu@ip-10-0-0-9:~$ cat ~/.ssh/id\_rsa.pub

ssh-rsa  ubuntu@ip-10-0-0-9

ubuntu@ip-10-0-0-9:~$ ^C

The last part was added into github SSH to establish the connection

Ensure the SSH key is set up correctly and can authenticate with GitHub:

ssh -T git@github.com

sometimes, after adding the ssh to github, when ‘git fetch’, still get error:

[t18155cb@login2 [csf3] ACIST]$ git fetch Username for 'https://github.com': BianChang Password for 'https://BianChang@github.com': fatal: Authentication failed for 'https://github.com/BianChang/ACIST.git/'

The error indicates that the repository's remote URL is set to use HTTPS instead of SSH. Since you've already added your SSH key, you need to update the remote URL to use the SSH protocol.

Follow these steps to resolve the issue:

**1. Check the Current Remote URL**

Run this command to see the remote URL configuration:

bash

Copy code

git remote -v

You will see output like:

plaintext

Copy code

origin https://github.com/BianChang/ACIST.git (fetch)

origin https://github.com/BianChang/ACIST.git (push)

This confirms that the repository is using HTTPS.

**2. Update the Remote URL to Use SSH**

Update the remote URL to use SSH instead of HTTPS:

bash

Copy code

git remote set-url origin git@github.com:BianChang/ACIST.git

Now verify the change:

bash

Copy code

git remote -v

The output should now look like:

plaintext

Copy code

origin git@github.com:BianChang/ACIST.git (fetch)

origin git@github.com:BianChang/ACIST.git (push)

1. Create Image from docker file:

ubuntu@ip-10-0-0-9:~/chang/infrastructure$ cd docker/

ubuntu@ip-10-0-0-9:~/chang/infrastructure/docker$ docker build -f Dockerfile.spotlight\_base -t spotlight\_base:latest .

check available Images: docker image list

REPOSITORY TAG IMAGE ID CREATED SIZE

spotlight\_base latest 928562bfdd6f 34 seconds ago 9.48GB

1. Create a container from the image:  
   Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Create and run a new container from an image

ubuntu@ip-10-0-0-9:~/chang$ docker run -v .:/workspace --gpus all -it spotlight\_base:latest

root@9b4ae2a4bd9f:/workspace# ls

infrastructure

docker ps: check containers

1. When running time is long, use screen to run jobs:

In root (container):

1. screen
2. screen -ls (check screens)
3. screen -r screenname (reactivate screen)
4. Ctrl + A ++D detach the screen
5. Windows VS code connects AWS-EC2

Host EC2

  HostName ec2-13-40-72-34.eu-west-2.compute.amazonaws.com

  User ubuntu

  IdentityFile C:\Users\chang\.ssh\ml\_ec2\_jan\_2024.pem

Note that the pem security: advanced-remove all, disable inheritage, leave only one, the User of the key (enter username (e.g. chang) in the check name box)

Hugging face write token for spotlight: hf\_UMZkjtuIKQizKDtsbBAdkapPvgKjwJADdg

11/7/2024

1. Use git lfs to handle large file commits in GitHub repository:
   1. Install git lfs
   2. git lfs install to initialize
   3. git track large files..:
      1. git lfs track "\*.bin"
      2. git lfs track "\*.safetensors"
   4. commit
2. after create a new branch, when operating on a different machine/server, use: git fetch origin to fetch the new created branch; may required enter the passphrase (bian1219), then you can use git branch -r to see the remote branches, and use git branch -a to see all; use git checkout <branchname > to checkout

11/11/2024

1. Every time into the container, need to pip install . lymph node triage and grout
2. For feature extraction test: python /workspace/chang/lymph-node-triage/lymphtriage/preprocessing/feature\_extraction.py --dataset /workspace/data\_dev\_s3/dev\_dataset\_s3.csv --s3-bucket "spotlight-pathology-lymphnode-datasets" --filepath-col filepath --label-col diagnostic.category.id --artifacts-dir /workspace/data\_dev\_s3/slide-artefacts --logs /workspace/chang/tmp/feature\_extraction.out --artifacts-format png

11/15/2024

1. Feature\_extraction.py of the cohort 1\_2:

python /workspace/chang/lymph-node-triage/lymphtriage/preprocessing/feature\_extraction.py --dataset /workspace/cohort\_1\_2\_partition/cohorts\_1\_2\_database\_20240912\_145230.csv --s3-bucket "spotlight-pathology-lymphnode-datasets" --filepath-col file\_path --label-col lymphoma\_diagnostic\_category\_name --tile 224 --tile-stride 224 --artifacts-dir /workspace/chang/histoqc-slide-artifacts/manchester-2022/ --logs /workspace/chang/tmp/feature\_extraction\_virchow-cohort-1-2.out --artifacts-format png --extractor virchow --dont-extract-if-exists

11/20/2024

1. docker run -v .:/histoqc\_space --gpus all -it histotools/histoqc:master /bin/bash
2. python histoqc\_helper.py --output-dir /histoqc\_space/chang/tmp/histoqc\_Salford\_output --ini-file /histoqc\_space/chang/lymph-node-triage/histoqc\_config\_manchester.ini

11/27/2024

1. set up anew branch:

git branch temp\_vit

git checkout temp\_vit

git push -u origin temp\_vit