

DS-GA 1006 Capstone

Lab Session 1

Instructors: Julia Kempe
Mark Ho
Najoung Kim
Wenda Zhou

TA: Wenxin Zhang (wz2164@nyu.edu)

Homework 1

- Goal of HW1
 - Realize use of SSH key pair, SSH agent, SSH agent forwarding
 - Get set up on the nyugateway, Greene cluster, burst login node, GCP cloud burst instance using SSH
 - Connect Greene cluster to Github using SSH
 - Have basic knowledge of NFS (networked file storage)
 - Connect VsCode to burst instance for later remote development
- Due Sep 20 (Please take a look early)
- Feel free to post your questions on BrightSpace Discussions

SSH

```
[(base) wenxinzhang@WenxinZhangsMBP ~ % ls -lah ~/.ssh
total 40
drwx-----  8 wenxinzhang  staff   256B Sep 12 10:32 .
drwxr-x---+ 42 wenxinzhang  staff   1.3K Sep 12 10:36 ..
srw-----  1 wenxinzhang  staff    0B Sep 12 09:25 .wz2164@gw.hpc.nyu.edu:22
-rw-r--r--@  1 wenxinzhang  staff   814B Sep 12 10:36 config
-rw-----  1 wenxinzhang  staff   2.5K Sep 12 10:23 id_rsa
-rw-r--r--  1 wenxinzhang  staff   576B Sep 12 10:23 id_rsa.pub
-rw-----  1 wenxinzhang  staff   3.3K Sep 12 01:27 known_hosts
-rw-----  1 wenxinzhang  staff   2.7K Sep 12 01:27 known_hosts.old
```

```
[wz2164@log-3 ~]$ ls -lah ~/.ssh
total 11K
drwx-----  2 wz2164 wz2164  4.0K Sep  9 14:37 .
drwx----- 16 wz2164 wz2164  8.0K Sep 11 20:50 ..
-rw-----  1 wz2164 wz2164   854 Sep  9 10:38 authorized_keys
-rw-----  1 wz2164 wz2164  3.4K Sep  9 10:38 id_rsa
-rw-r--r--  1 wz2164 wz2164   750 Sep  9 10:16 id_rsa.pub
-rw-r--r--  1 wz2164 wz2164  1.1K Sep 11 20:08 known_hosts
```

- SSH (Secure Shell Protocol)
 - A cryptographic network communication protocol for operating network services over an unsecured network
 - Widely used in cloud services, network environments, file transfer tools and etc.
- Configure SSH to multiplex connections through files found in ~/.ssh folder
 - `ls -lah ~/.ssh` (list out files in ~/.ssh folder)
 - `vim ~/.ssh/config` (use “`esc + :wq`” to save and quit)
- SSH key is the secure access credential used in SSH. It uses key pairs for identity authentication
- ~/.ssh/id_rsa.pub (stores your public key)
 - Copy your public key to the remote server for authentication using “`ssh-copy-id greene`”
 - Once the server receives your public key and considers it trustworthy, the server marks the key as authorized in `authorized_keys`
- ~/.ssh/id_rsa (stores your private key)
 - The possession of this private key is proof of user’s identity. Store it carefully.

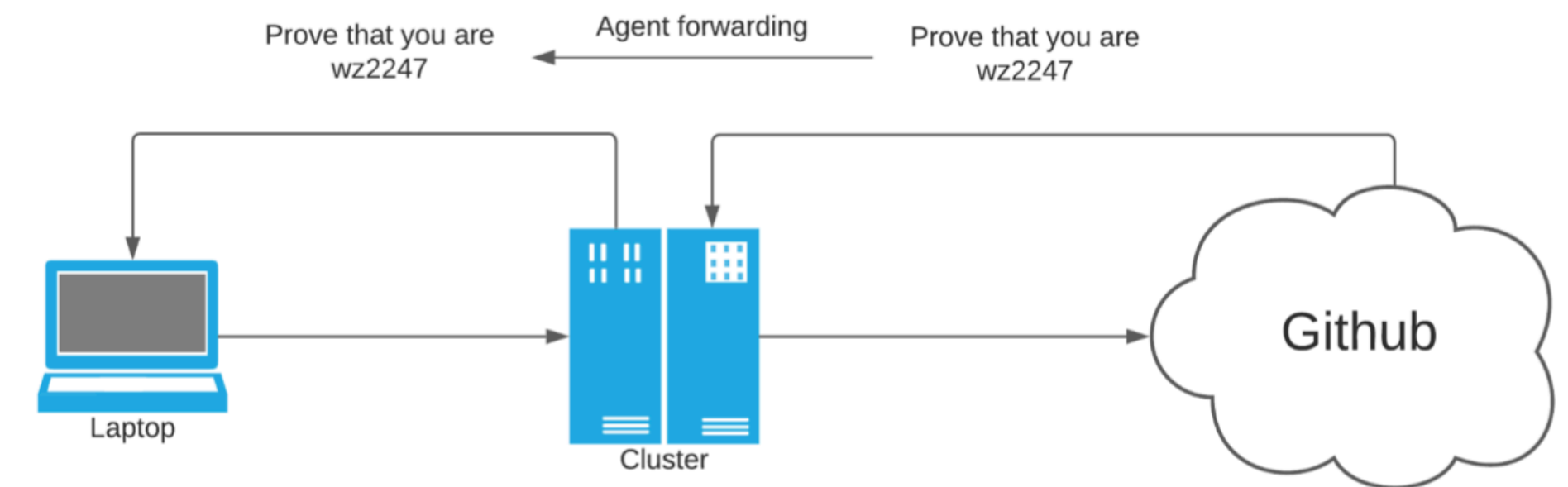
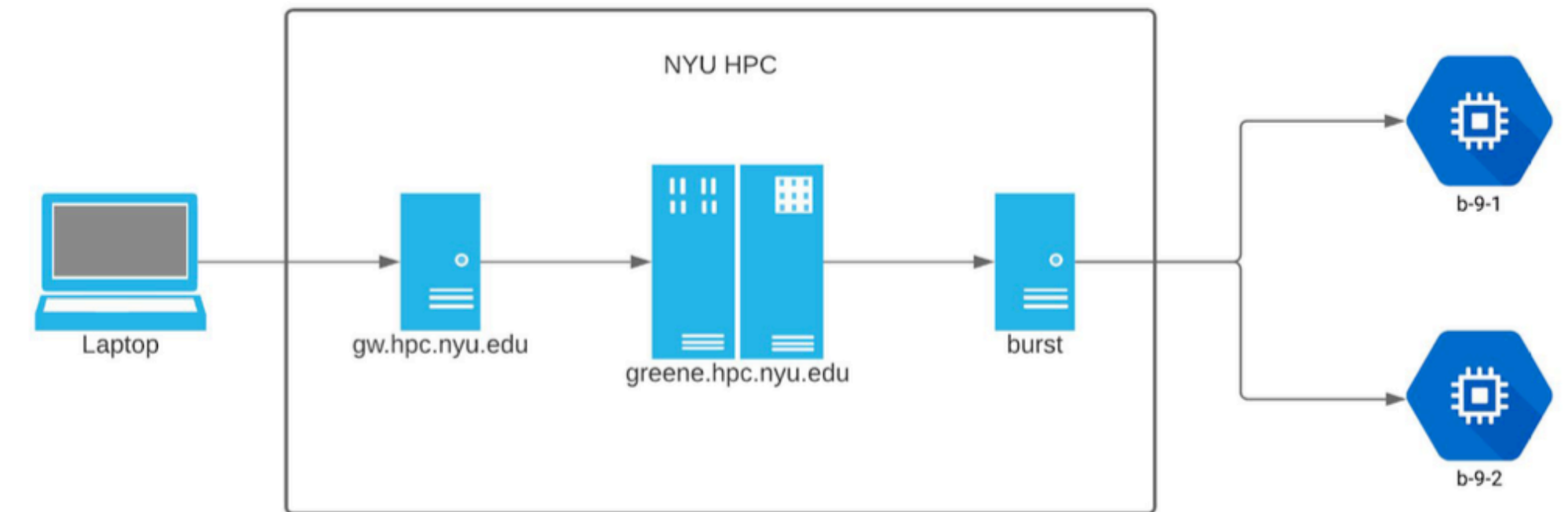
SSH Agent & Agent Forwarding

- **SSH Agent**

- Every time you need to use your private key, SSH will ask you for a passphrase. This can be troublesome!
- SSH agent helps keep your private key in memory, so that you can log into the servers securely without a passphrase prompt.
- Add your SSH key to the agent using `ssh-add .ssh/id_rsa`.

- **SSH Agent Forwarding (~/.ssh/config: ForwardAgent yes)**

- We can ask SSH to forward the authentication through agent forwarding
- Since you don't want to store your private keys on the server, SSH agent can help forward the identification request to your laptop. Likewise, Your laptop will answer the question and send the response back to the cluster, and to the Github.



SSH Setup

Configure your `~/.ssh/config` to enable passwordless login to the NYU HPC Greene cluster, as well as key-based authentication on your github account. To ensure that your setup functions properly, run the following commands:

1. `ssh greene`
2. `hostname`
3. `ssh -T git@github.com`

There should be no authentication prompts which are required, and you should see Github responding with your git username. An example session is displayed in listing 1. Please include a screenshot of your terminal with the relevant command and their output displayed.

```
[WENDA-SURFACE]$ ssh greene
```

[illegible]

Last login: Wed Sep 7 02:08:01 2022 from 10.47.6.5

```
[wz2247@log-2 ~]$ hostname
```

log-2.hpc.nyu.edu

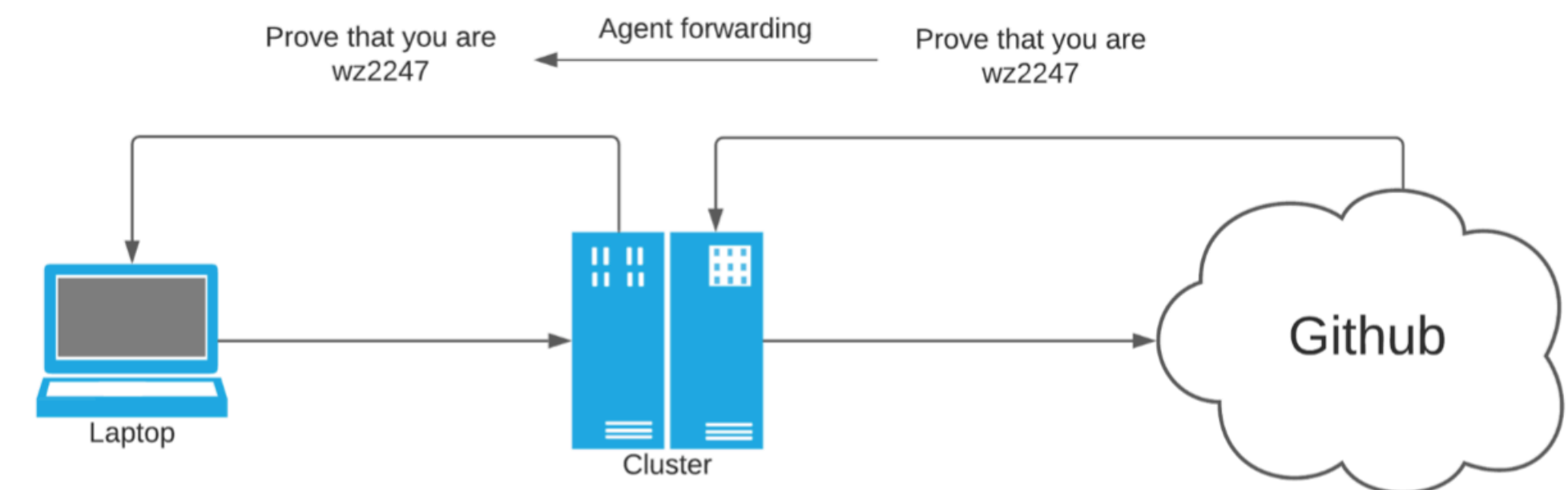
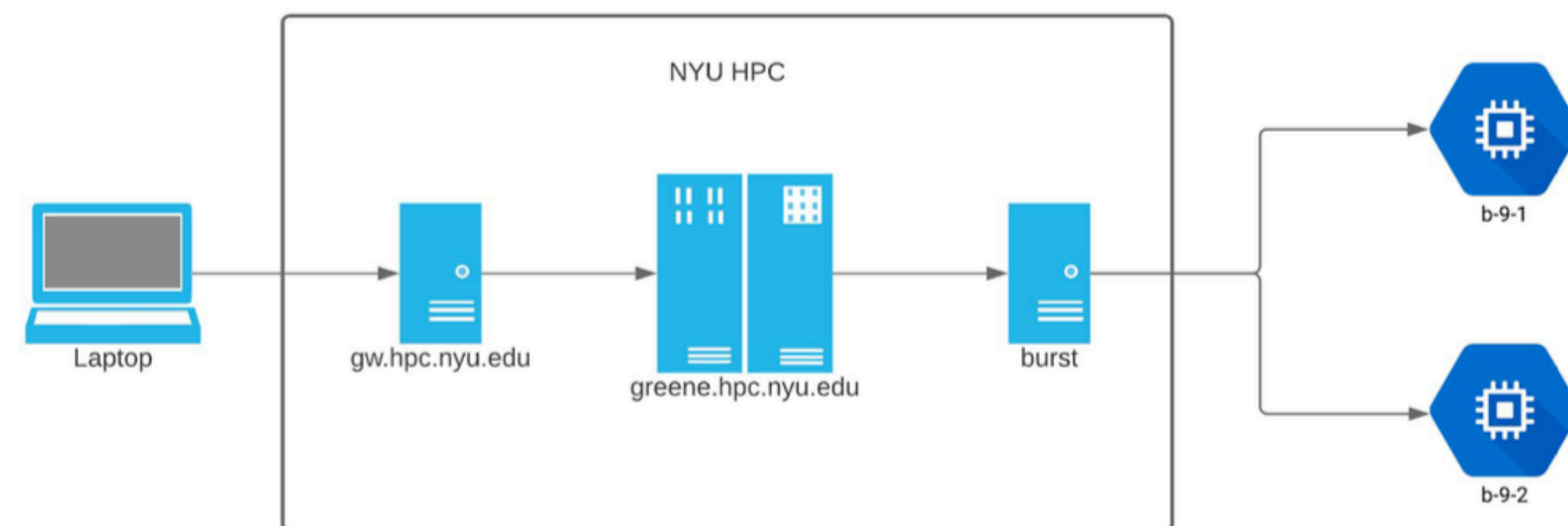
```
[wz2247@log-2 ~]$ ssh -T git@github.com
```

```
Hi wendazhou! You've successfully authenticated, but GitHub does not provide shell access.
```

```
[wz2247@log-2 ~]$ exit
```

[logout](#)

Connection to greene.hpc.nyu.edu closed.



SSH Setup

Configure your ~/.ssh/config to enable passwordless login to the NYU HPC Greene cluster, as well as key-based authentication on your GitHub account.

```
Host nyugateway
  User wz2164
  HostName gw.hpc.nyu.edu
  ForwardAgent yes
  ControlPath ~/.ssh/.*r@%h:%p
  ControlMaster auto
  ControlPersist yes

Host greene
  User wz2164
  HostName greene.hpc.nyu.edu
  ForwardAgent yes
  ProxyJump nyugateway
  StrictHostKeyChecking=No
  UserKnownHostsFile=/dev/null

Host greeneburst
  User wz2164
  HostName log-burst.hpc.nyu.edu
  ForwardAgent yes
  ProxyJump greene

Host burstinstance
  User wz2164
  HostName b-2-1 # Need to change as needed
  ForwardAgent yes
  ProxyJump greeneburst
```

1. Edit SSH configuration on VSCode to multiplex connections
2. Check whether your public/private key pair exists
 - (id_*.pub, id_*) `ls -la ~/.ssh`
 - (Generate a new pair) `ssh-keygen -t ed25519 -C "your_email@example.com"`
3. Add your SSH key to the agent
 - `ssh-add ~/.ssh/id_rsa`
4. Add your public key on Greene to enable login without password
 - `ssh-copy-id greene`
5. After login to Greene, add your public key on Github
 - <https://docs.github.com/en/authentication/connecting-to-github-with-ssh>
6. Check the connection:
 - On your laptop: `ssh greene/ greeneburst`
 - On Greene: `hostname`
 - On Greene: `ssh -T git@github.com` [test ssh authentication for Github]

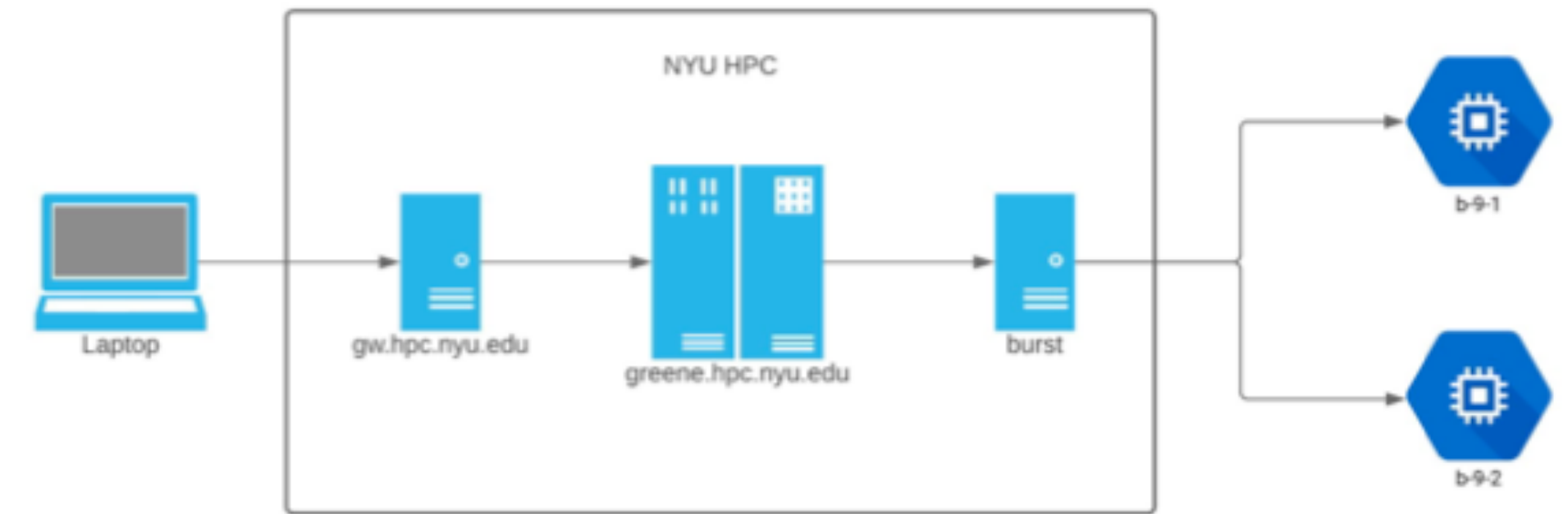
GCP cloud bursting setup

Request a new instance on the GCP cloud bursting using the `c2s16p` partition (CPU-only). Modify your `.ssh/config` to be able to directly connect to your allocated GCP instance.

Run the command below and include a screenshot of the output in your terminal (note: I have included a redirection of `stderr` to remove some clutter, remove `2>/dev/null` and see what it looks like without the redirection).

```
ssh -T burstinstance 'cat /scratch/wz2247/cds-bootcamp/homework/url.txt' 2>/dev/null
```

GCP cloud bursting setup



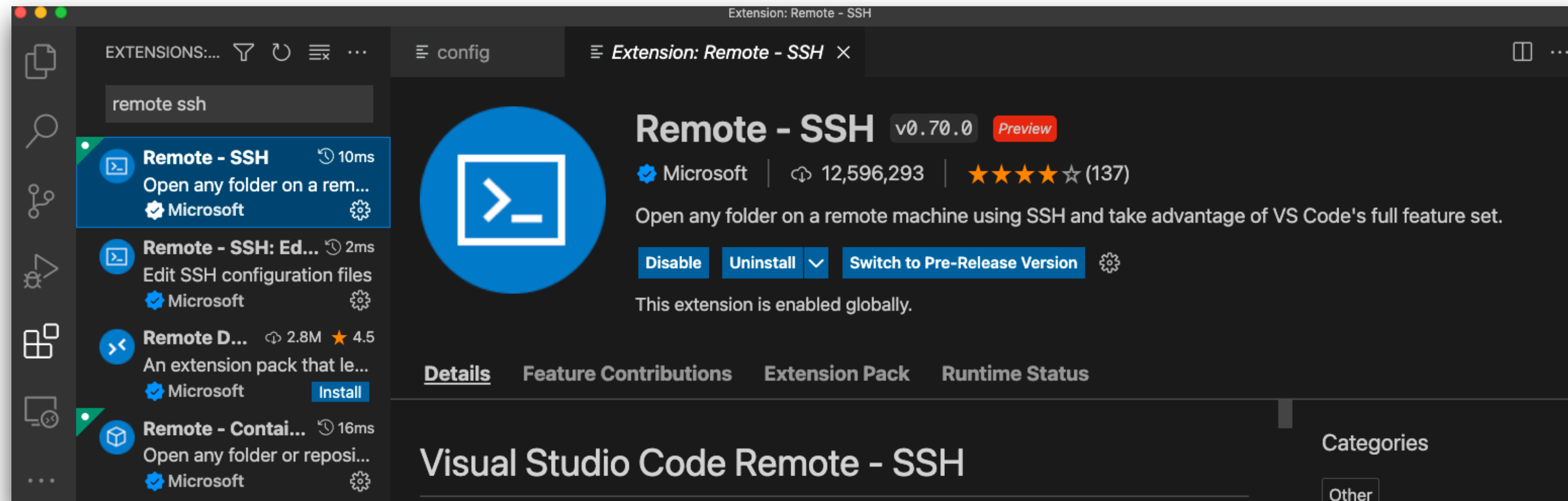
1. ssh to the Burst login node from your laptop
 - “ssh greeneburst”
2. Start an interactive job/ Request a new GCP cloud bursting instance using Slurm
 - “srun --account=ds_ga_1006_001-2022fa --partition=c2s16p --time=2:00:00 --pty /bin/bash”
3. Check the allocated node & Modify the ~/.ssh/config on your laptop to point to your allocated server and
 - “squeue -u \$USER” & Hostname b-2-1
4. ssh to the GCP cloud bursting instance *from your laptop* (“.ssh/config” is local)
 - First, you need to setup. Send your public key from greene to your burst instance!
 - In the shell into the instance, run “scp greene-dtn:~/.ssh/authorized_keys ~/.ssh/”
 - The Greene node and the burst login node use a NFS (networked file storage) system and can see the same files
 - When we ran “ssh-copy-id greene” in previous steps, SSH modified the authorized_keys file on Greene (and `burst` sees the change) to include your public key.
 - The GCP machines also use a NFS, but a different one, which is stored within google cloud.
 - All instances of GCP machines do see the same files. But they don't see the same files as burst or Greene.
 - Keep your GCP instance open and connect to the instance from Terminal or VsCode Remote SSH

```
[[wz2164@b-2-3 ~]$ cat /scratch/wz2247/cds-bootcamp/homework/url.txt  
https://www.youtube.com/watch?v=dQw4w9WgXcQ
```

```
xinZhangsMBP ~ % ssh -T burstinstance 'cat /scratch/wz2247/cds-bootcamp/homework/url.txt' 2>/dev/null  
https://www.youtube.com/watch?v=dQw4w9WgXcQ
```


VsCode remote

Request a new instance (or re-use the same instance from the previous section) on the GCP cloud bursting platform using the c2s16p partition. Install VSCode and the remote extension for VSCode. Connect VSCode to your GCP instance. Confirm that your VSCode is indeed running on the remote instance by opening the terminal in VSCode (**View > Terminal**) and querying the current host name (using the 'hostname' command). Include a screenshot of your VScode window.



How to Maintain the Access to GCP Instance?

- “The connection to the burstinstance is pretty unstable, it often cuts itself off with an error of: broken pipeline, how can we keep the connection to the burstinstance?”
- Solution: Create a dummy sbatch script that just calls sleep and run that non-interactively if you have lots of issues. But **REMEMBER to cancel your job** after you are done working to preserve computational resources!!!

```
wenxinzhang — wz2164@log-burst:~...  
#!/bin/bash  
#  
#SBATCH --job-name=sleep  
#SBATCH --output=output.txt  
#SBATCH --account=ds_ga_1006_001-2022fa  
#SBATCH --partition=c2s16p  
#  
#SBATCH --time=5:00:00  
#SBATCH --ntasks=1  
#SBATCH --cpus-per-task=1  
#SBATCH --mem-per-cpu=100  
  
sleep 1h
```

- Create a dummy sbatch script and save the changes: `vim sleep.sh`
- Run the sbatch non-interactively: `sbatch sleep.sh`
- Check your \$jobid and allocated burst instance: `queue -u wz2164`
- Cancel the job after you are done working: `scancel $jobid`