CSCI4180: Tutorial 5

Amazon Elastic Compute Cloud (EC2) Guide

Huancheng Puyang

Department of Computer Science & Engineering
The Chinese University of Hong Kong

2022.10.19

Who am I

- Name: Huancheng Puyang
- Program: PhD of CSE
- Research Interest: Network & Storage
- Email: hcpuyang22@cse.cuhk.edu.hk
- Piazza: https://piazza.com/cuhk.edu.hk/fall2022/csci4180/home

Outline

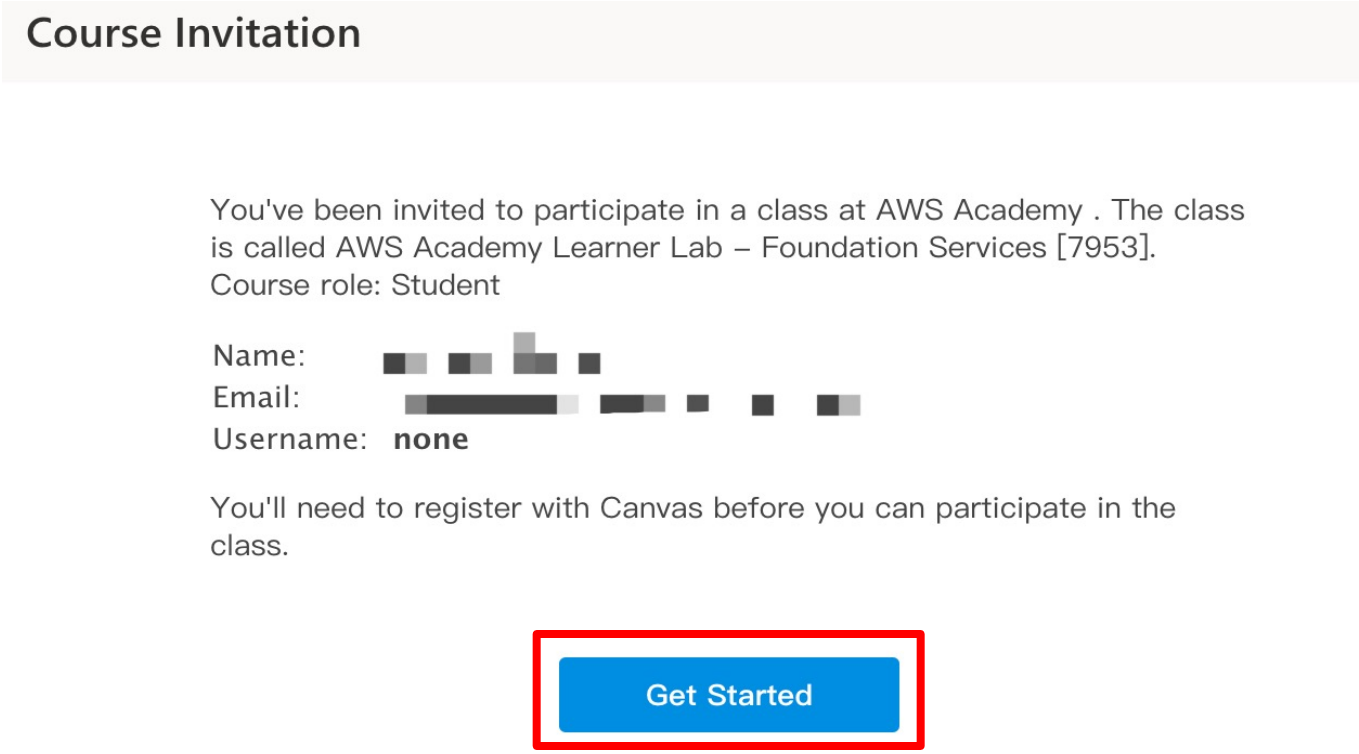
- Apply Amazon Web Services (AWS) Educate Grant
- Launch Amazon EC2 instance
- Configure Hadoop cluster on Amazon EC2
- Run MapReduce on Amazon EC2

Outline

- Apply Amazon Web Services (AWS) Educate Grant
- Launch Amazon EC2 instance
- Configure Hadoop cluster on Amazon EC2
- Run MapReduce on Amazon EC2

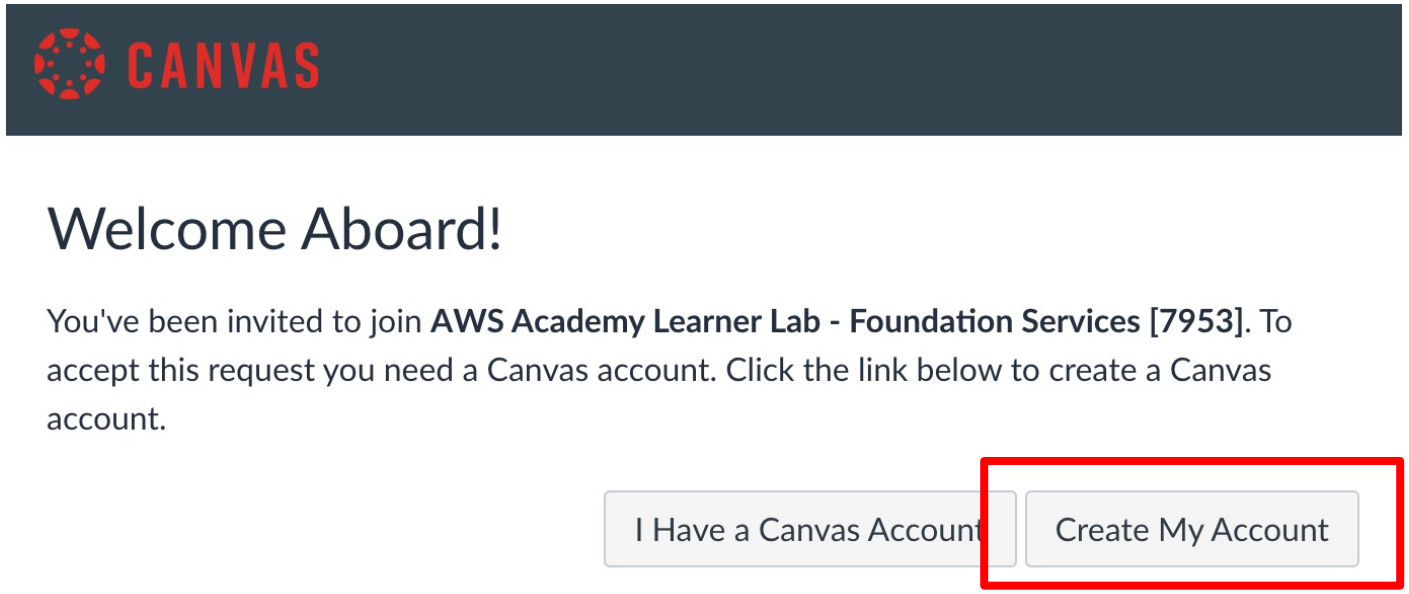
Amazon Web Services (AWS) Educate Grant

- Step 1. Receive an invitation link
  - Click ”Get Started”
  - **Ask on Piazza if no invitation email by tomorrow**



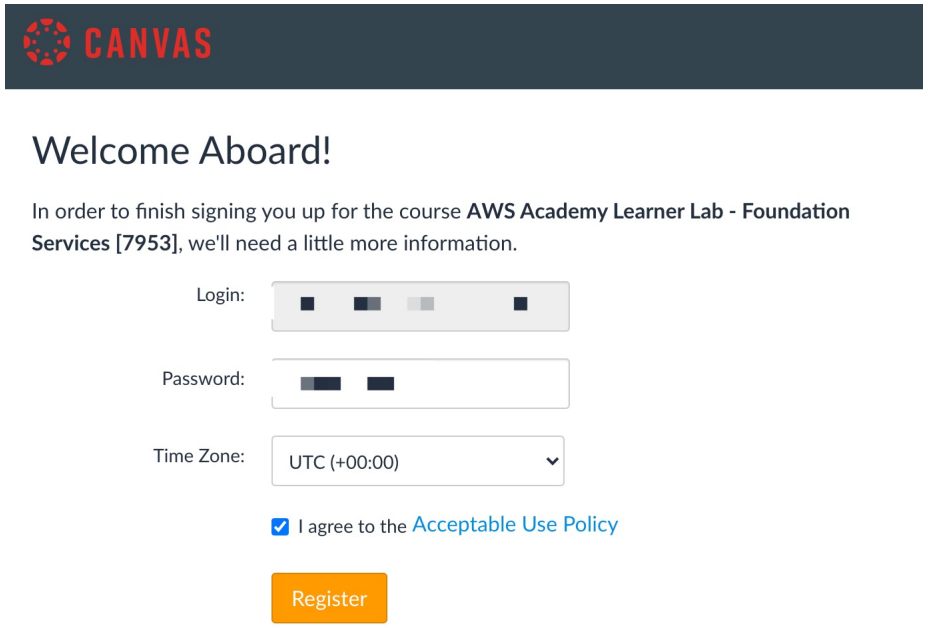
Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 1. Receive an invitation link
  - Click “Create My Account”
  - **Canvas account is different from AWS account (if you have)**



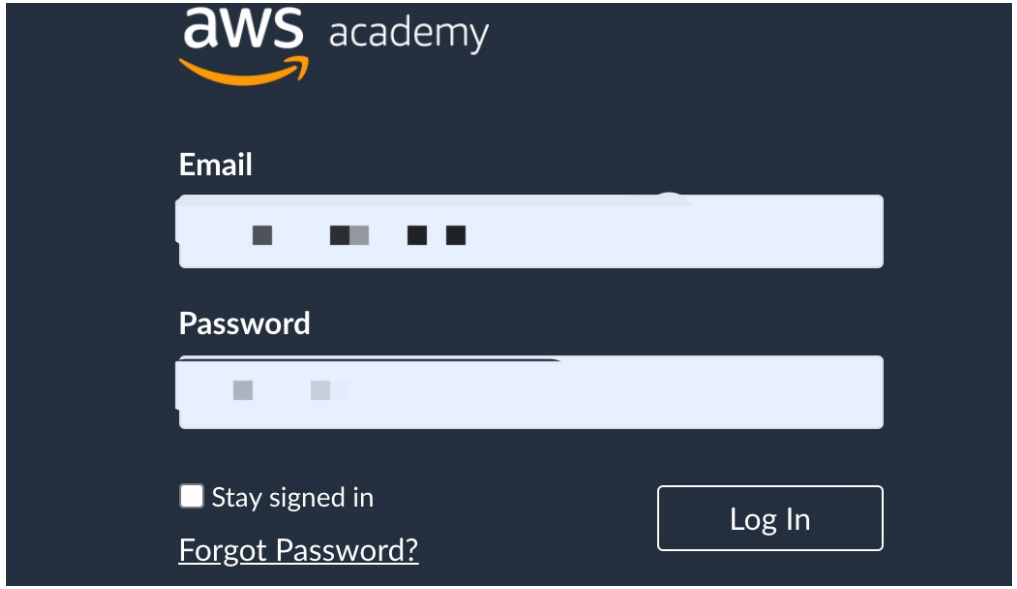
Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 1. Receive an invitation link
  - Register the course
  - **Get 100$ credits during course**



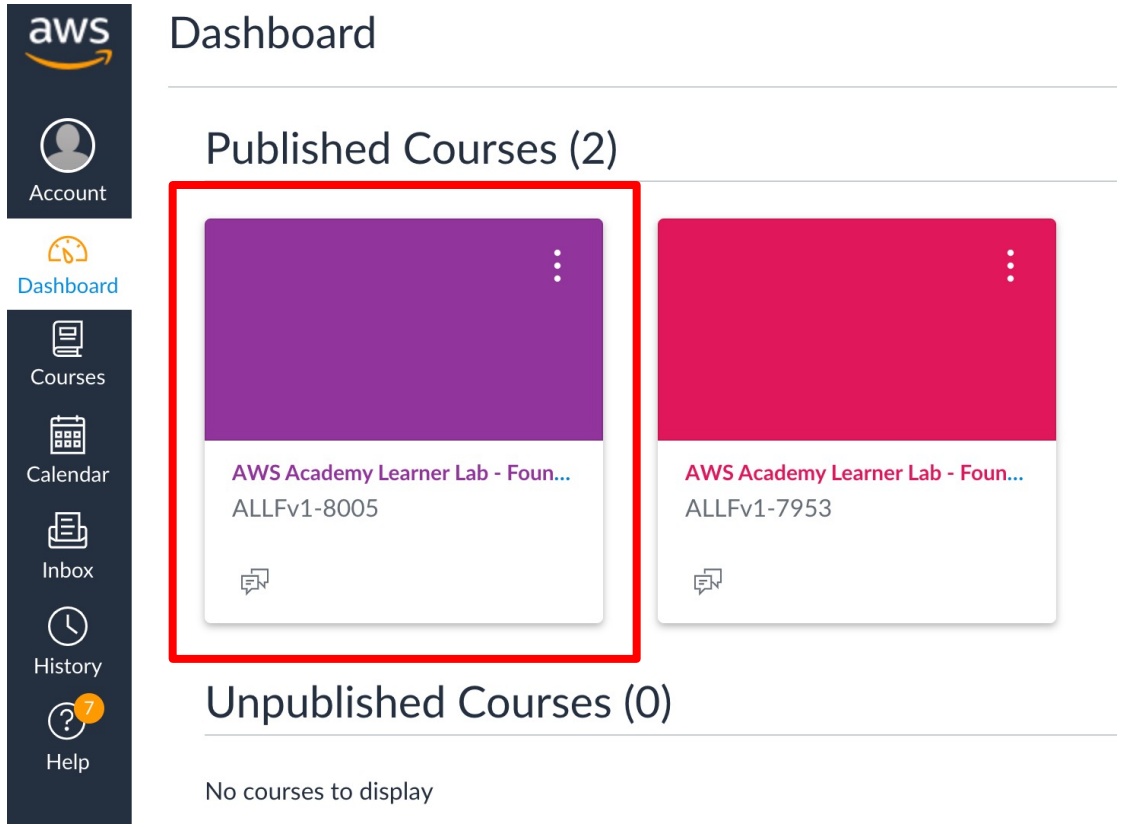
Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 2. Open lab terminal
  - Log in: https://www.awsacademy.com/LMS_Login
  - Click “Student Login” and input email & passsword



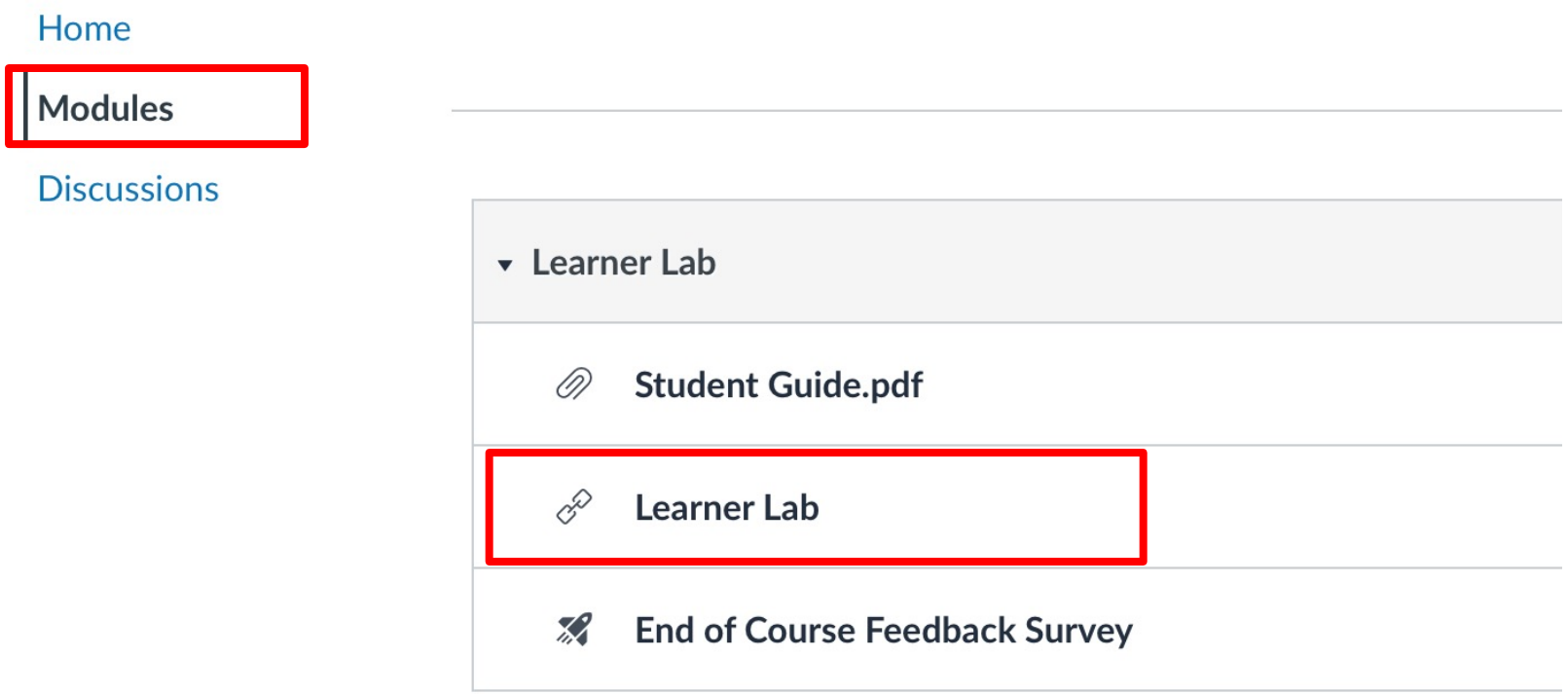
Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 2. Open lab terminal
  - Choose Learner Lab (You will have only one choice)



Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 2. Open lab terminal
  - Choose “Modules”
  - Choose “Learner Lab”

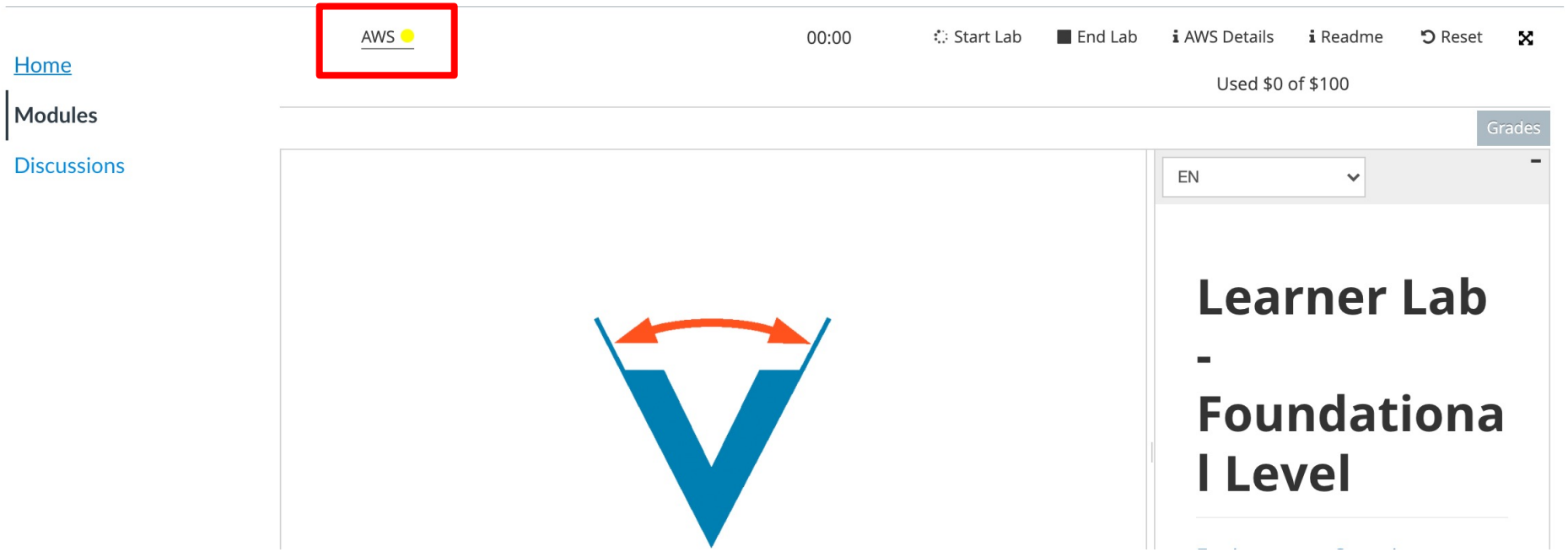


Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 2. Open lab terminal
  - The terminal is used for Canvas account management instead of virtual machine (will **not be used**)
  - The status of AWS is red: lab session is not started
  - Click “Start Lab”

Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 2. Open lab terminal
  - Lab session is preparing for your (as well as your EC2 instances if any)
  - Wait the status of AWS converting from yellow to green

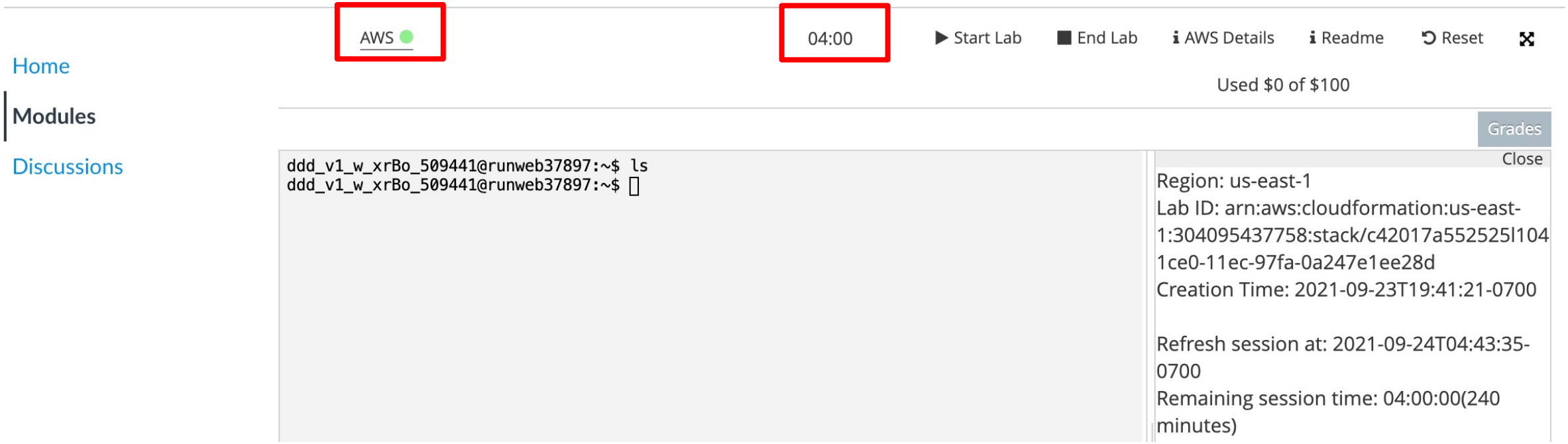


Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 3. Open AWS management console
  - The amount of used credits
    - The credit record is not updated in real-time (around every 8 hours)
    - **You have only 100$ credits which will not be regranted (keep economical)**

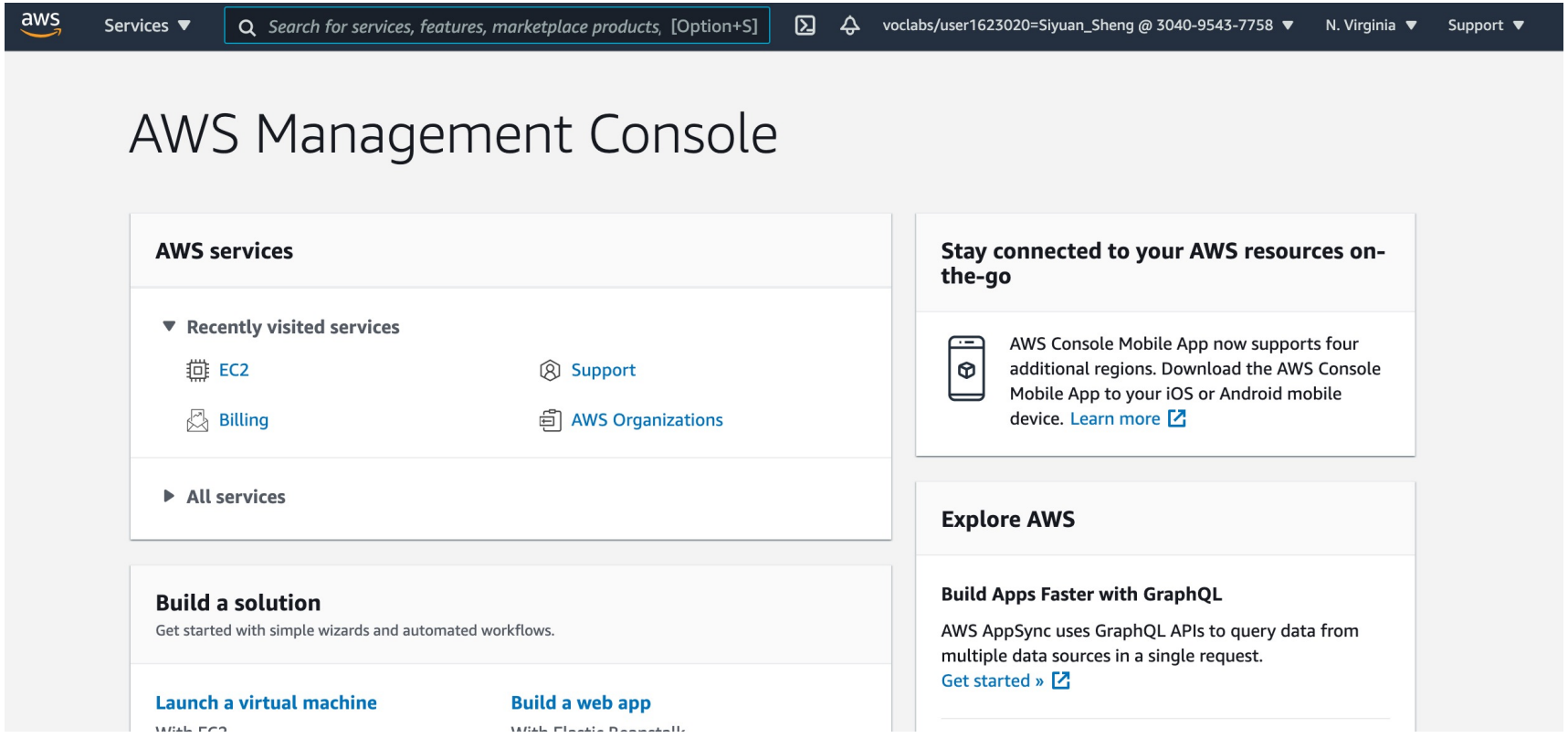
Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 3. Open AWS management console
  - After AWS turns green, click ”AWS”
  - Each session stops after 4 hours (choose “Start Lab” if session is stopped)



Amazon Web Services (AWS) Educate Grant (Cont.)

- Step 3. Open AWS management console
  - After clicking “AWS”, get AWS management console
  - You can launch and use EC2 instances here (detailed later)



Amazon Web Services (AWS) Educate Grant (Cont.)

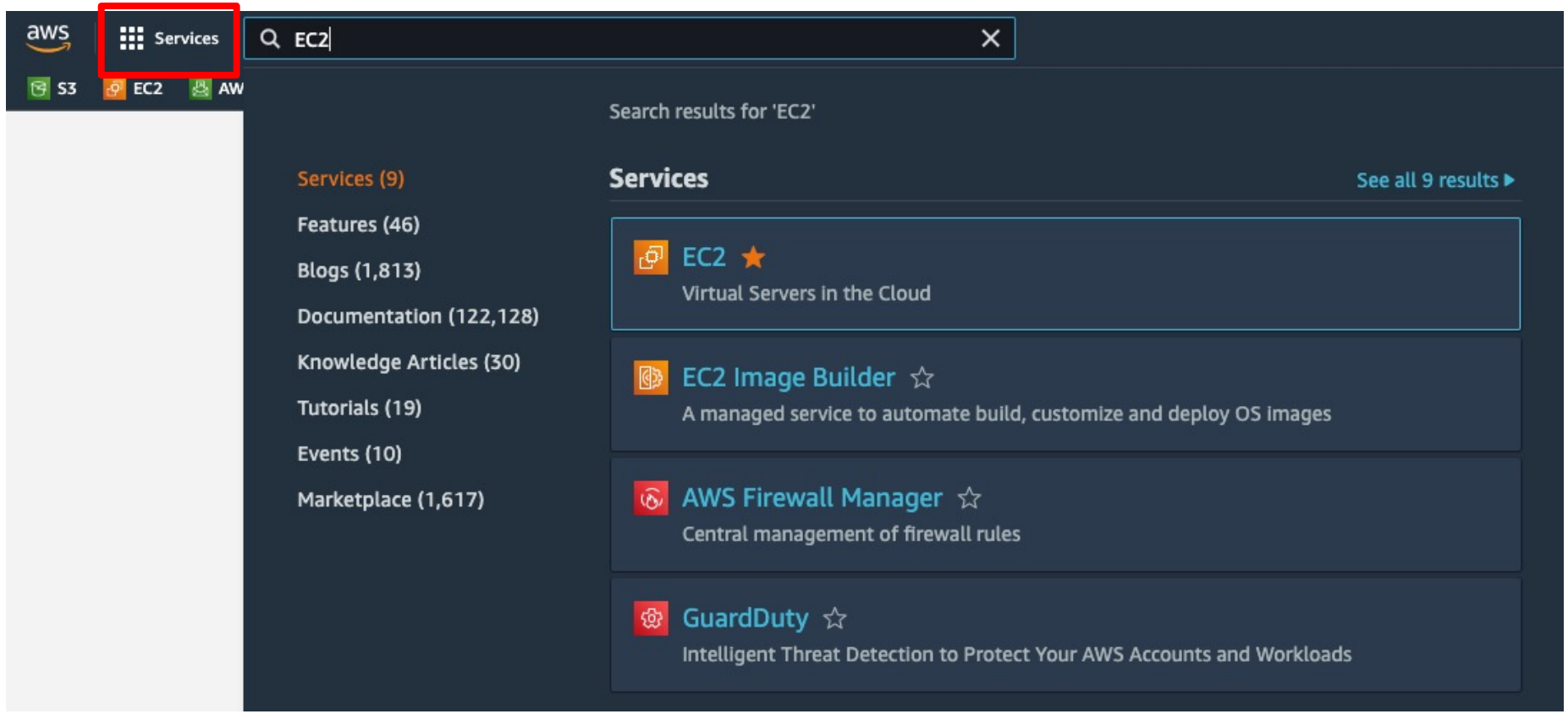
- Step 4. Close lab session
  - Click “End Lab” if you are not using your EC2 instances to **AVOID CHARGE**
  - Also close management console (do it after launching and using, **not now**)

Outline

- Apply Amazon Web Services (AWS) Educate Grant
- Launch Amazon EC2 instance
- Configure Hadoop cluster on Amazon EC2
- Run MapReduce on Amazon EC2

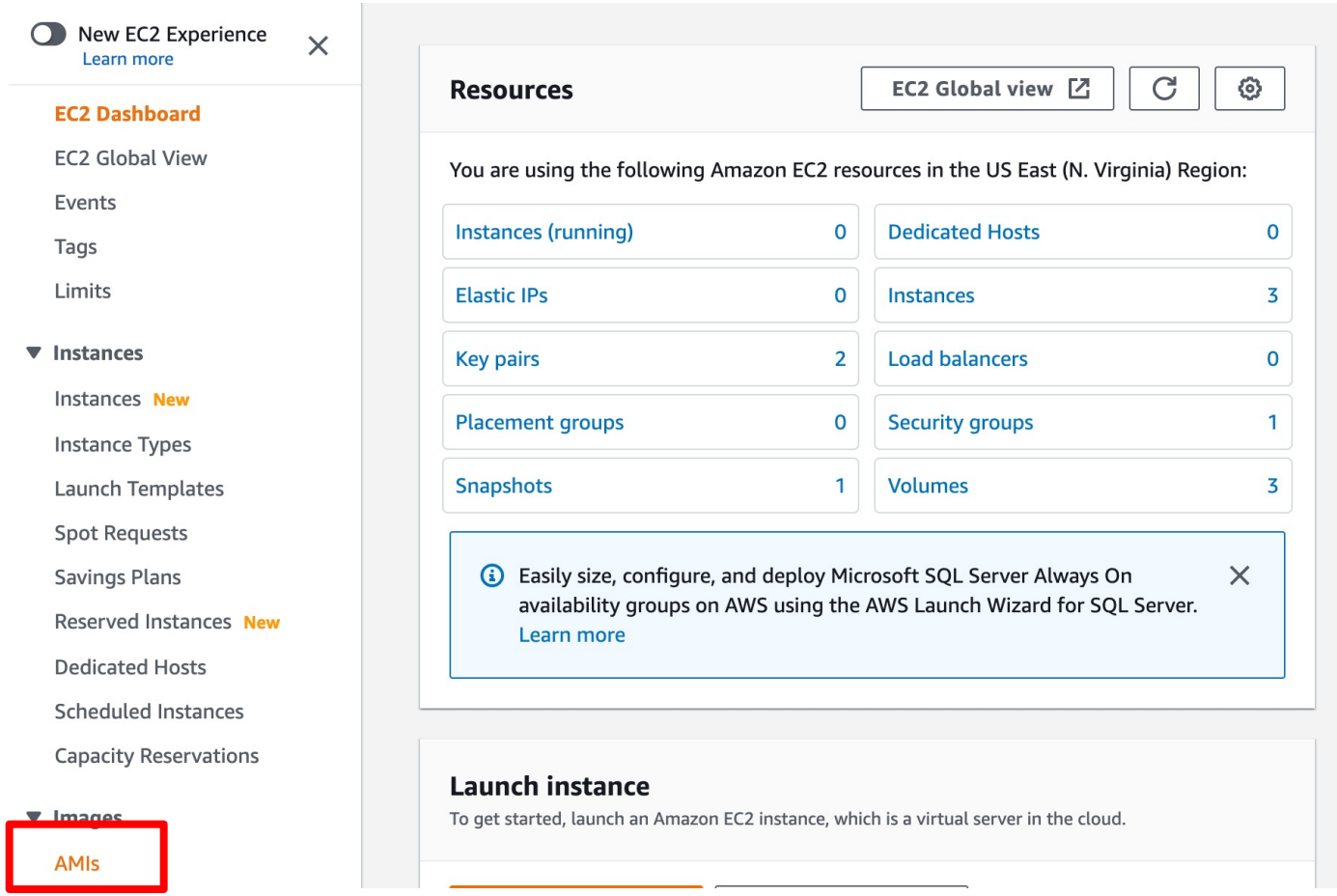
Launch Amazon EC2 instance

- Create an instance
  - Choose “Services” ➔ “EC2” in the management console



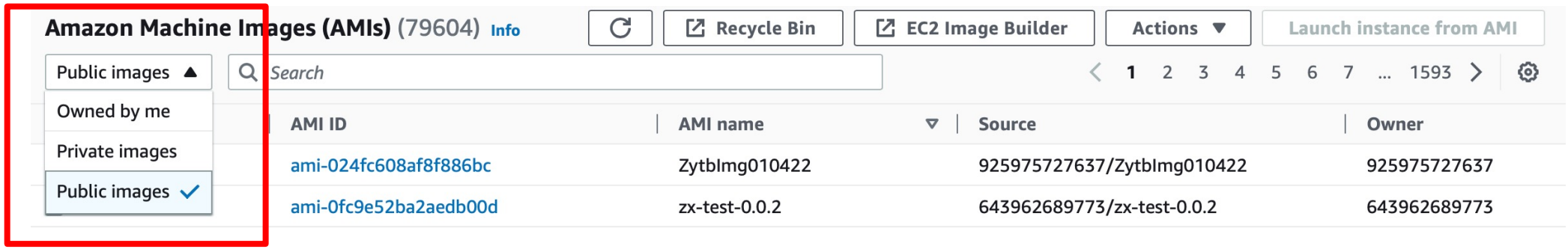
Launch Amazon EC2 instance (Cont.)

- Create an instance
  - Click ”AMIs” from the left bar



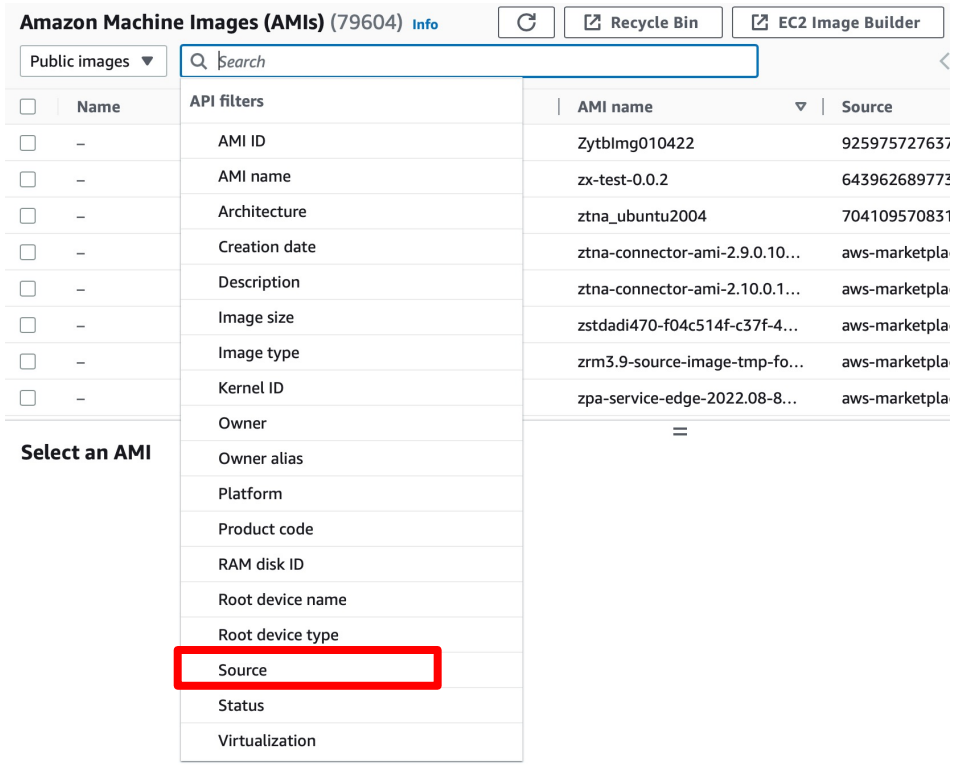
Launch Amazon EC2 instance (Cont.)

- Create an instance
  - Choose ”Public Images”



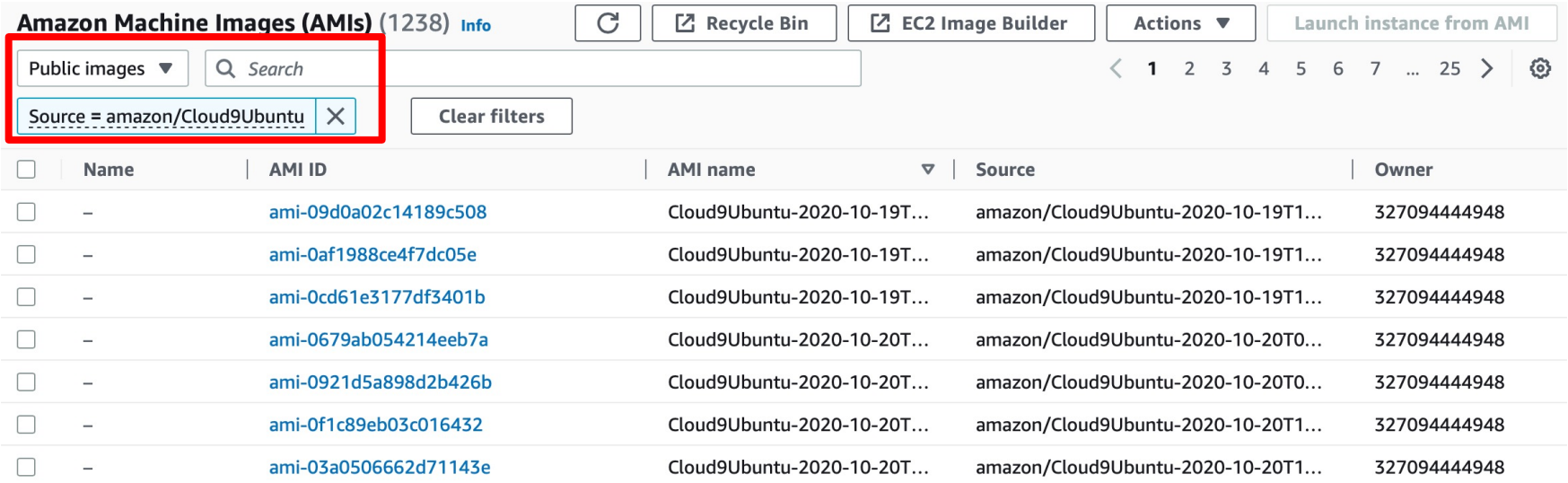
Launch Amazon EC2 instance (Cont.)

- Create an instance
  - Click the search bar and choose “Source”



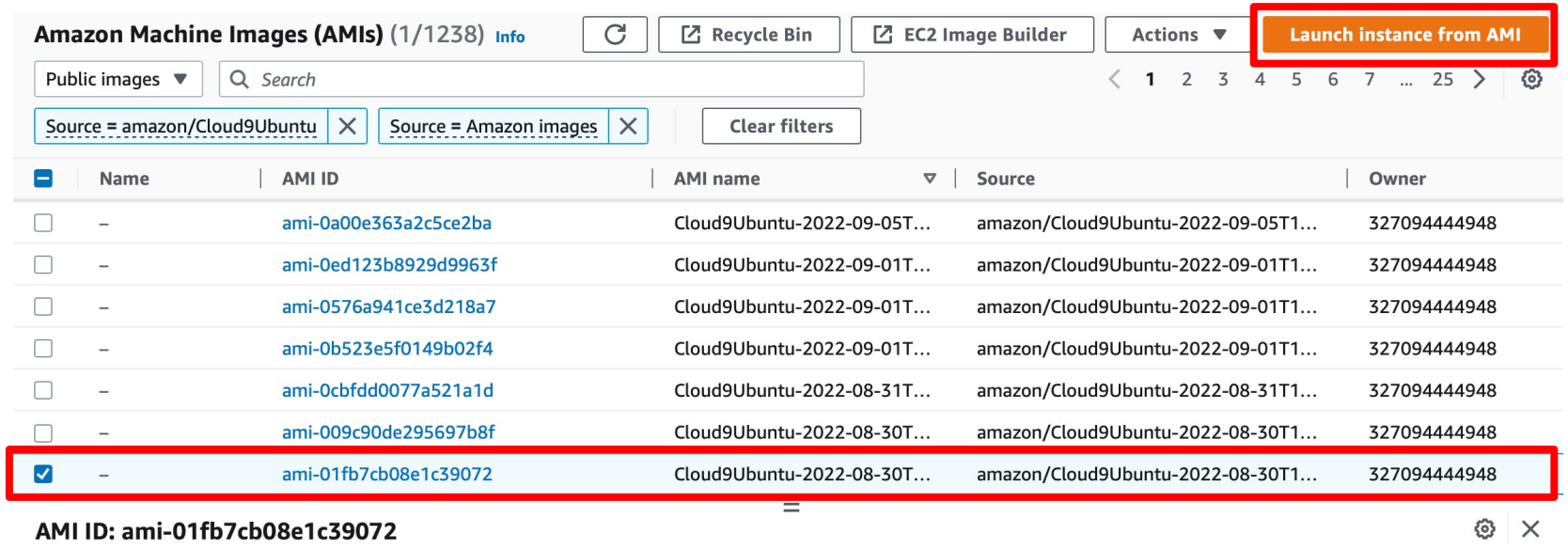
Launch Amazon EC2 instance (Cont.)

- Create an instance
  - Input “amazon/Cloud9Ubuntu” in the search bar for Source



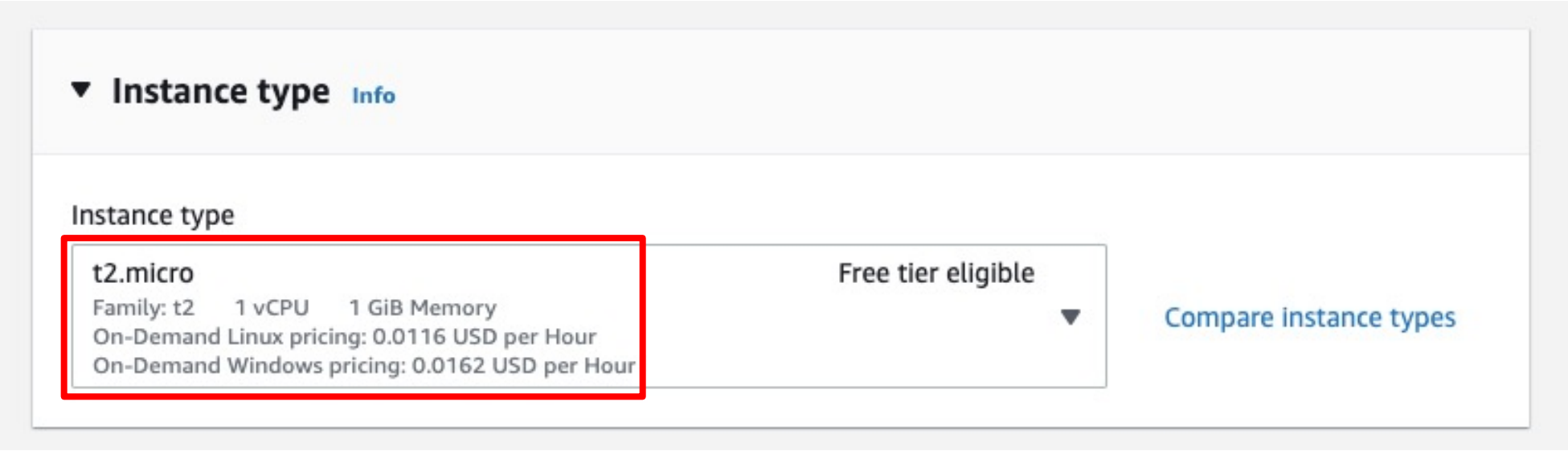
Launch Amazon EC2 instance (Cont.)

- Create an instance
  - Select “Cloud9Ubuntu-2022-08-30T11-46” and click “Launch instance from AMI”



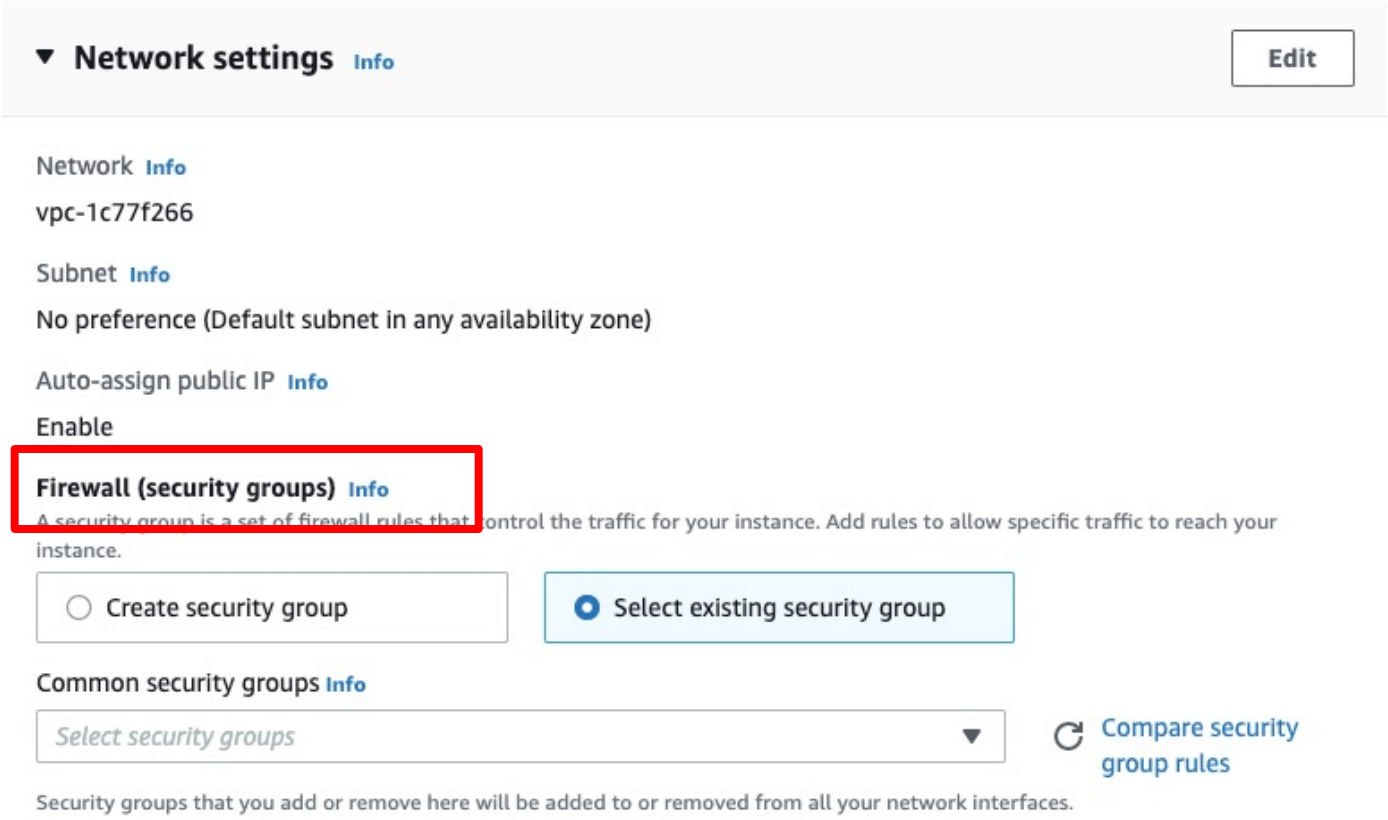
Launch Amazon EC2 instance (Cont.)

- Choose an Amazon Machine Image (AMI)
  - Choose an instance type
    - Click: “t2.micro”



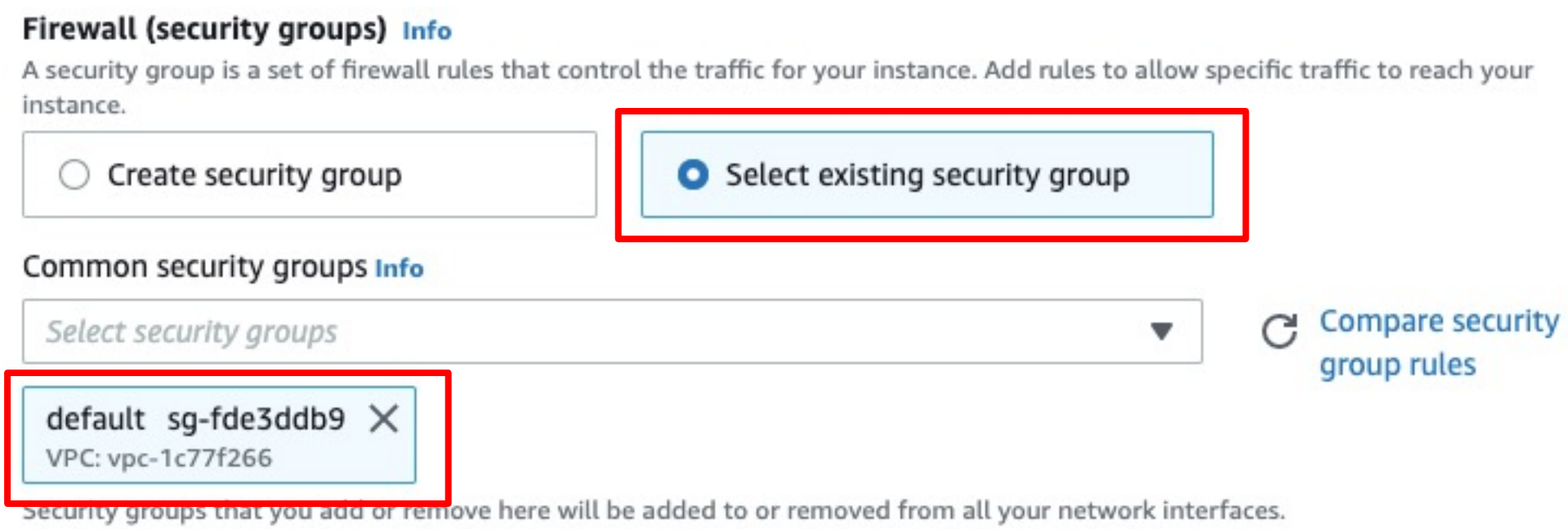
Launch Amazon EC2 instance (Cont.)

- Edit security settings and choose default security group
  - Click: “Edit security groups”



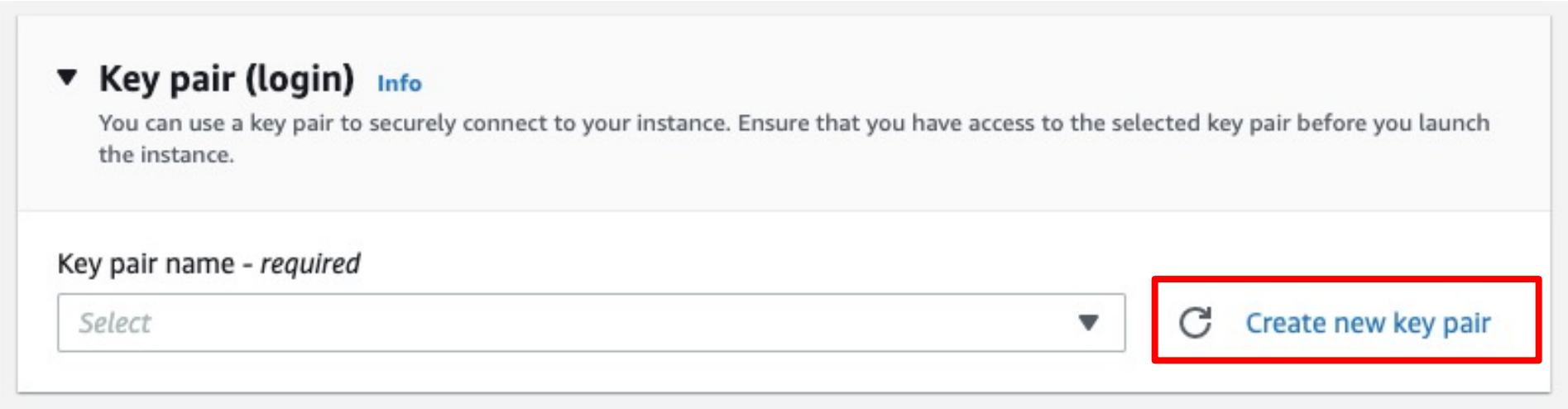
Launch Amazon EC2 instance (Cont.)

- Edit security settings and choose default security group
  - Click: “select an existing security group” → “security group ID”



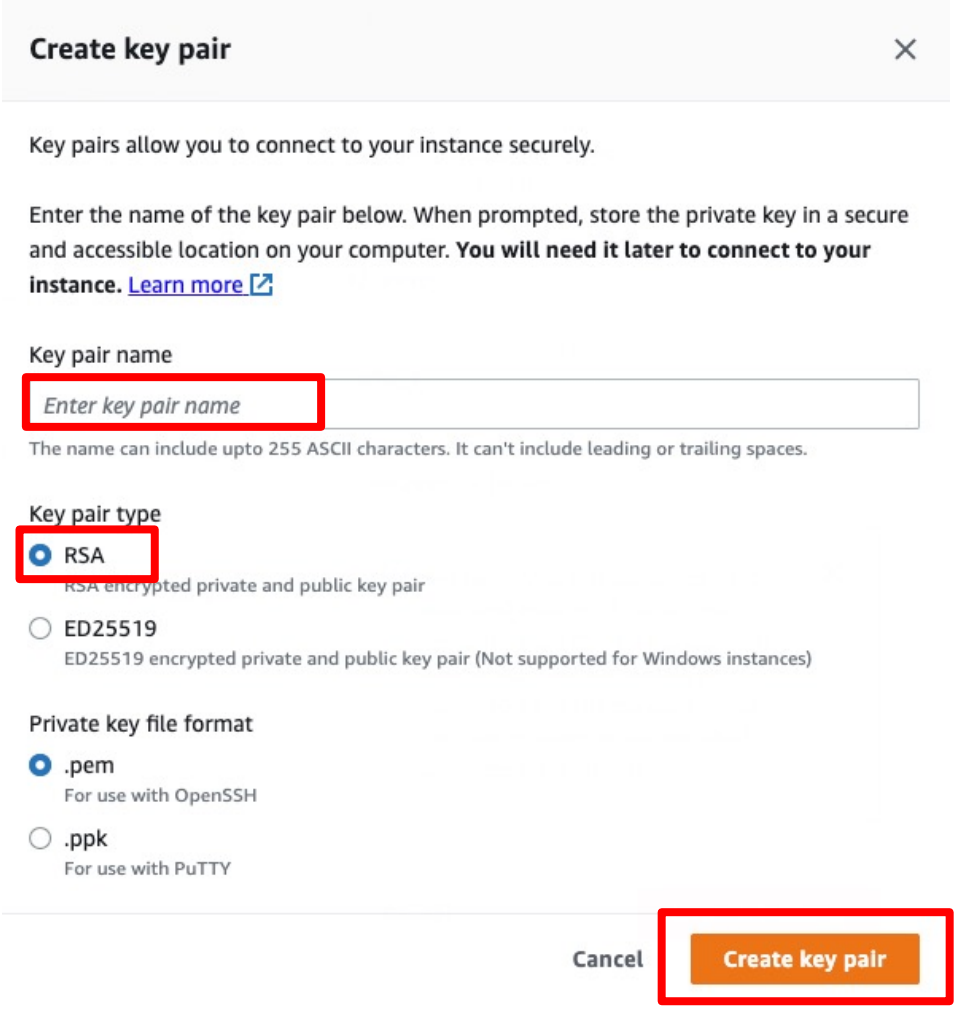
Launch Amazon EC2 instance (Cont.)

- Edit the access key
  - Select “Create new key pair”



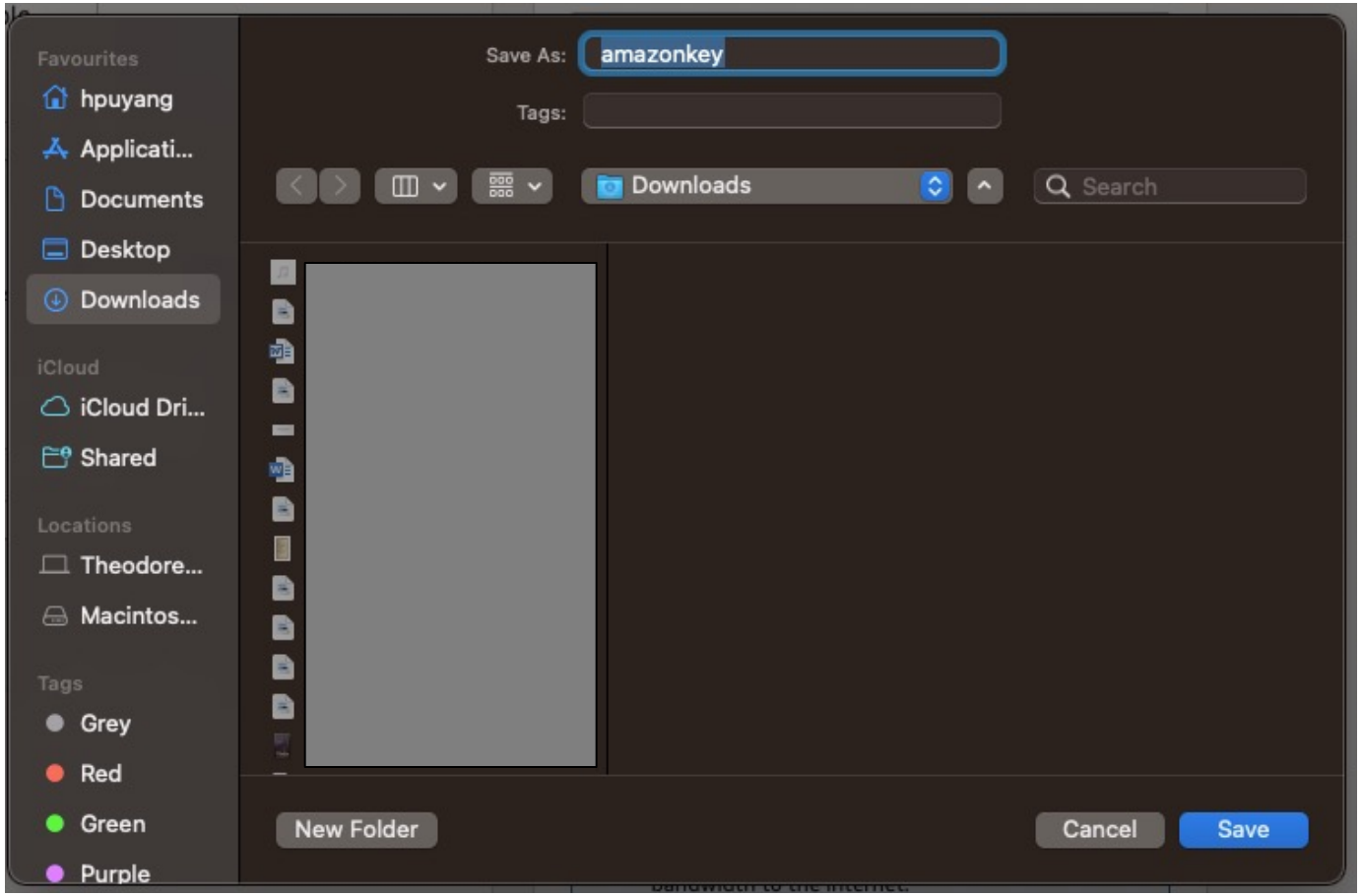
Launch Amazon EC2 instance (Cont.)

- Edit the access key
  - Select “Create a new key pair”
  - Input key name, e.g., “amazonkey”
  - Select ”RSA”
  - Click “Create key pair”



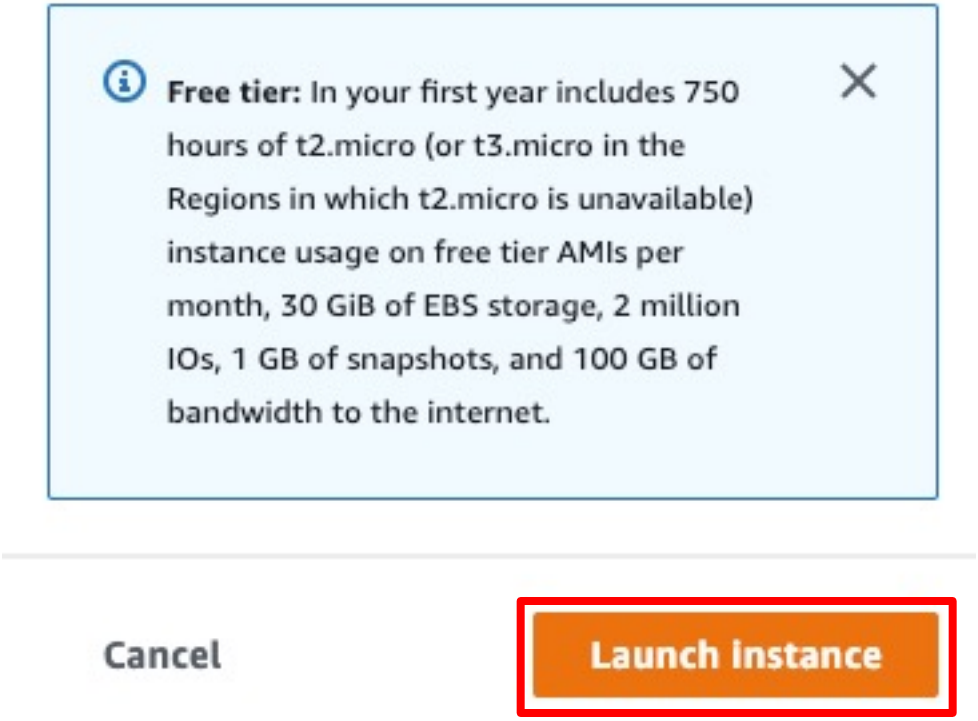
Launch Amazon EC2 instance (Cont.)

- Edit the access key
  - Save the key to path where you can use for the following steps.



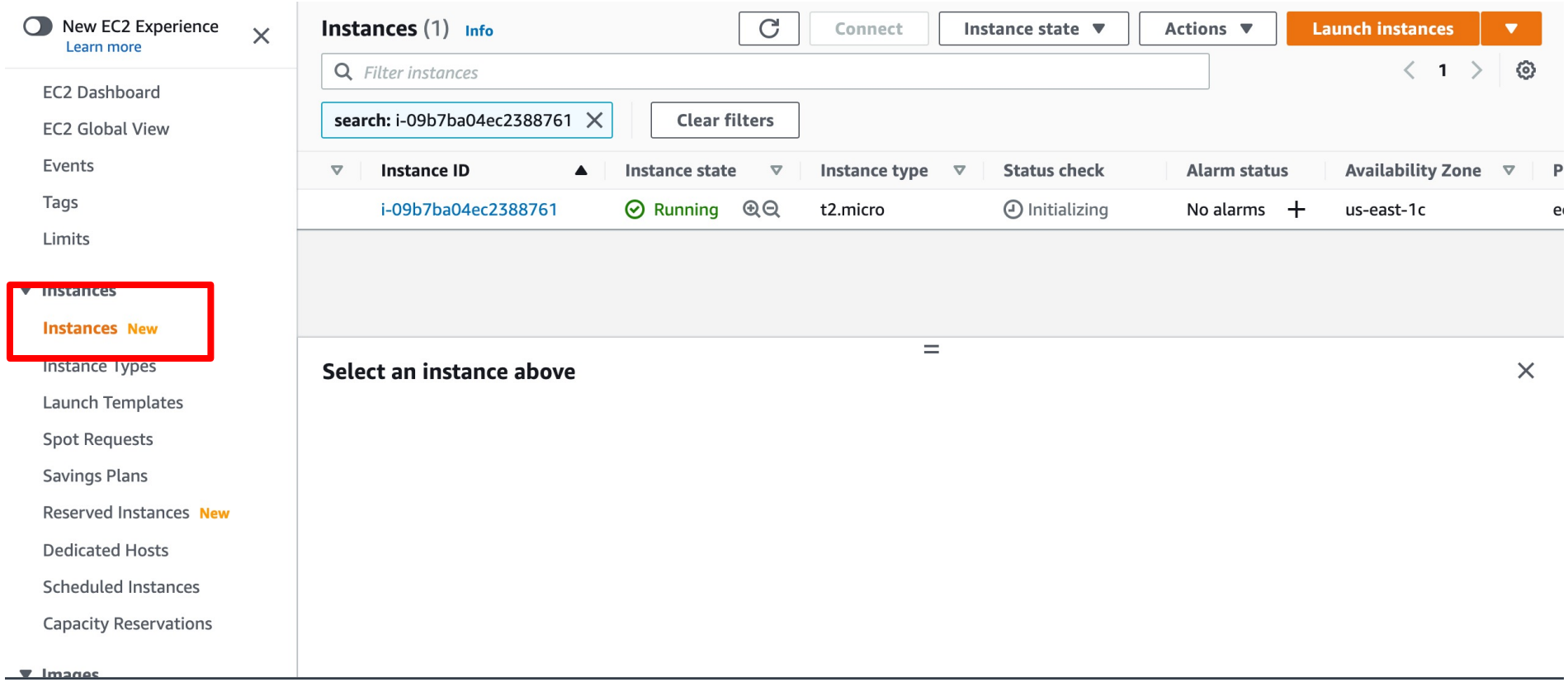
Launch Amazon EC2 instance (Cont.)

- Launch instance



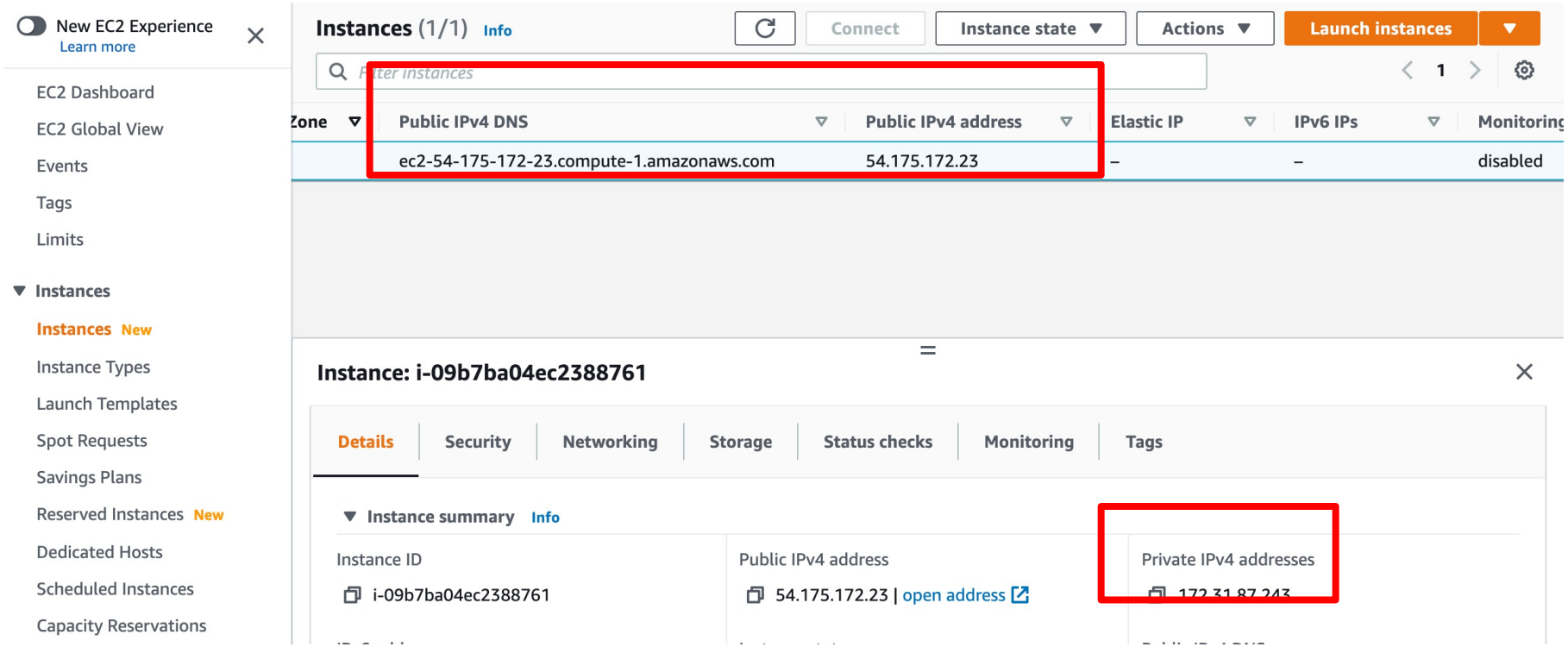
Launch Amazon EC2 instance (Cont.)

- View your Amazon EC2 instance
  - In AWS management console, choose “Instances” from the left bar



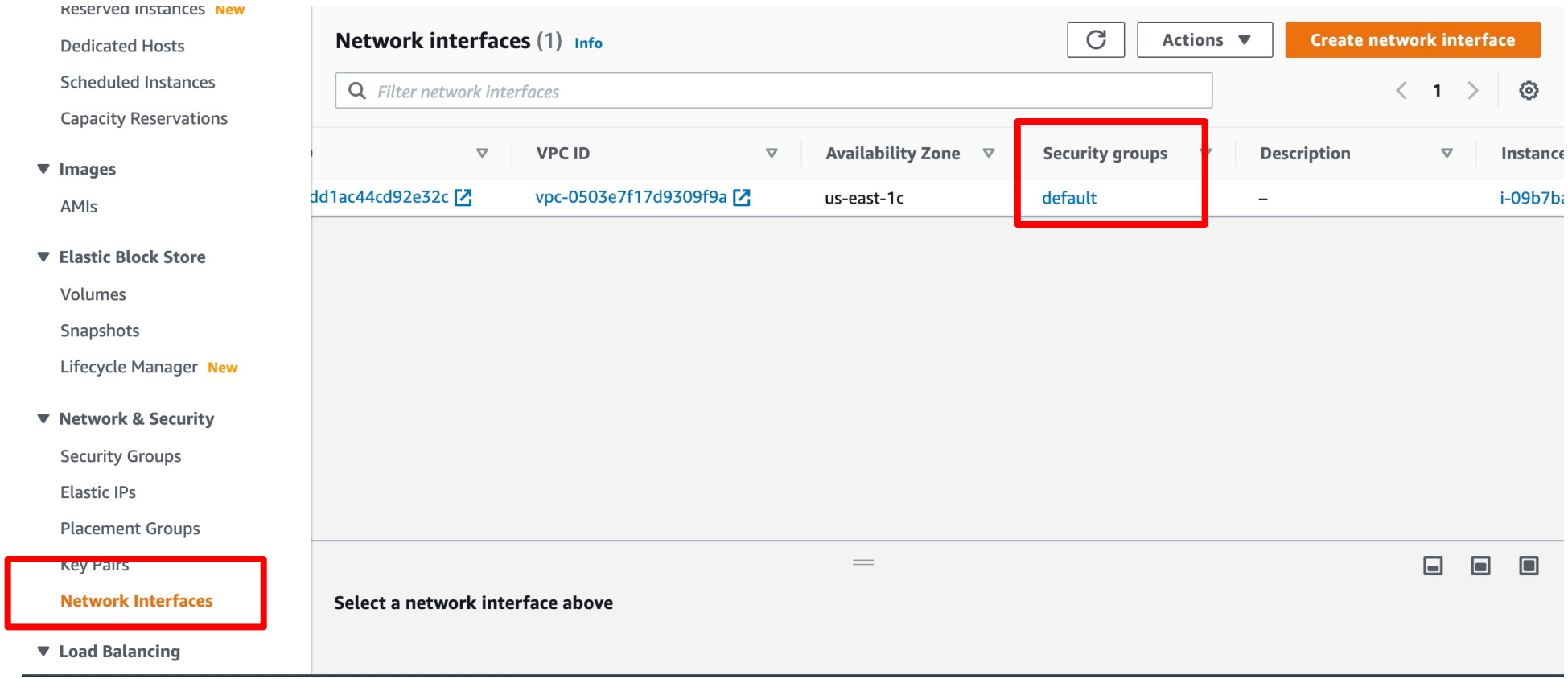
Launch Amazon EC2 instance (Cont.)

- View your Amazon EC2 instance
  - We will use **public DNS** to connect to EC2 instance
    - Each time when you restart your vm, **the public DNS and public IPv4 is changed**
    - **But the private IPv4 is static** (we will use it to configure Hadoop cluster)



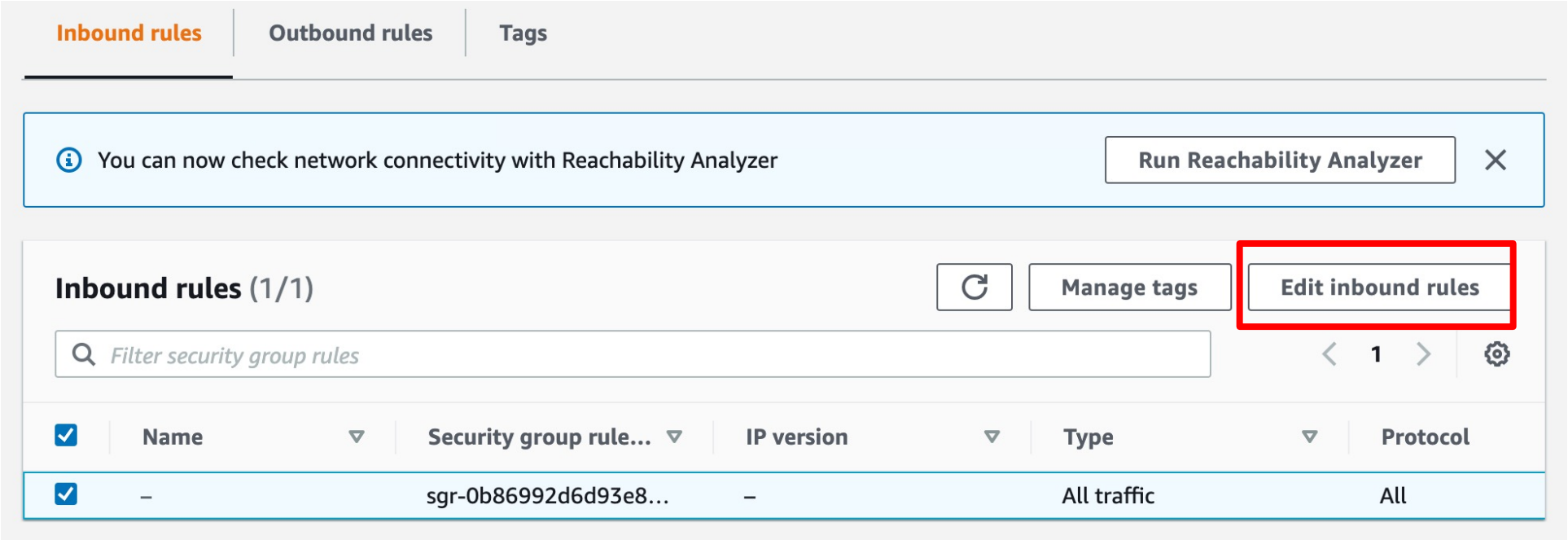
Launch Amazon EC2 instance (Cont.)

- Configure security group
  - Choose “Network Interfaces” and click “default” under security group



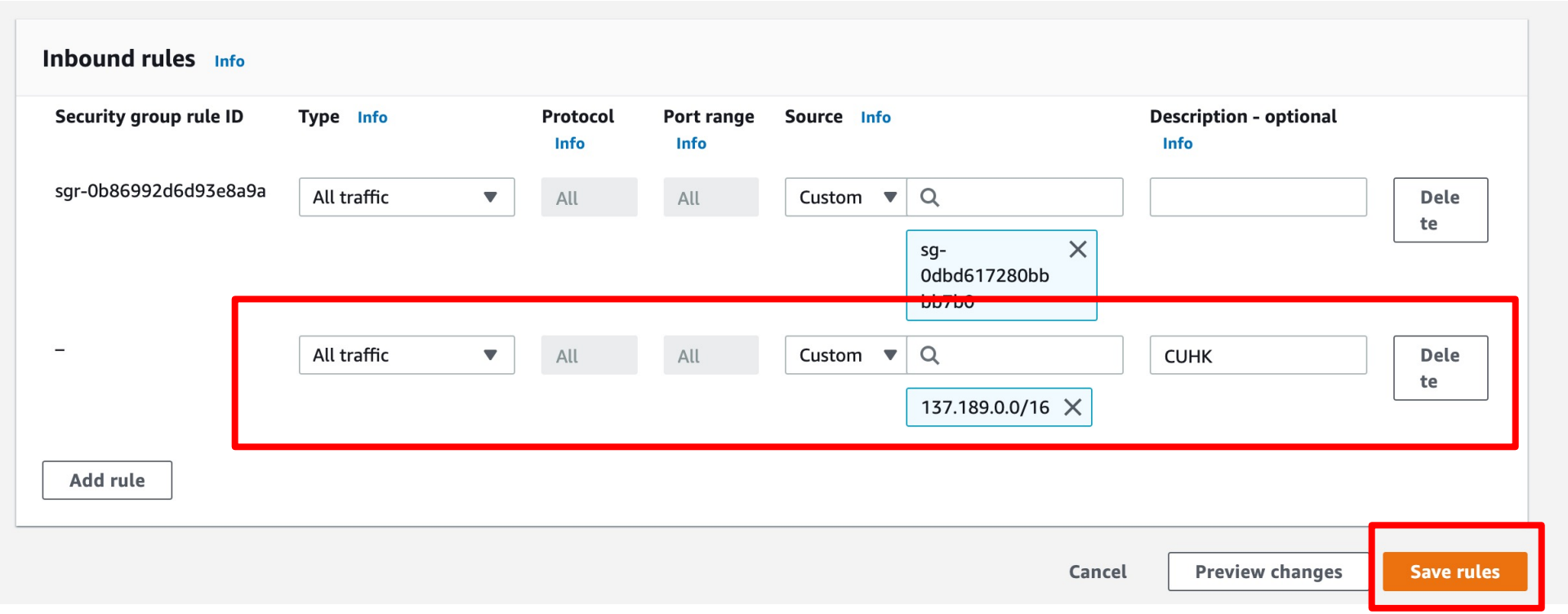
Launch Amazon EC2 instance (Cont.)

- Configure security group
  - Click “Edit inbound rules” to control the traffic to your instance
    - We allow the source IP from CUHK **(137.189.0.0/16)** to prevent DDoS attack.
    - We also allow the source IP from the private IP of each VM to configure Hadoop (detailed later)



Launch Amazon EC2 instance (Cont.)

- Configure security group
  - Add rule for the source IP from CUHK **(137.189.0.0/16)**
  - Click “Save Rules” (we add the source IP of our VMs later)



Launch Amazon EC2 instance (Cont.)

- Access to Amazon EC2 instance
  - Use “chmod” to change access permission of private key

```
$ chmod 0400 /{your_path}/amazonkey.cer
```

    - “/{your_path}/amazonkey.cer”: the path of your key downloaded previously
  - Configure ~/.ssh/config **if you are in CSE network**.

    (Change to *your* vm public DNS, and need to *reconfigure* after *restart* instance)

```
Host ec2-18-221-66-100.us-east-2.compute.amazonaws.com
ProxyCommand nc -x socks.cse.cuhk.edu.hk:1080 %h %p
```

  - Note
    - Delete “~/.ssh/config” when you are not in cse network
    - The default user of your VM is “**ubuntu**”

Launch Amazon EC2 instance (Cont.)

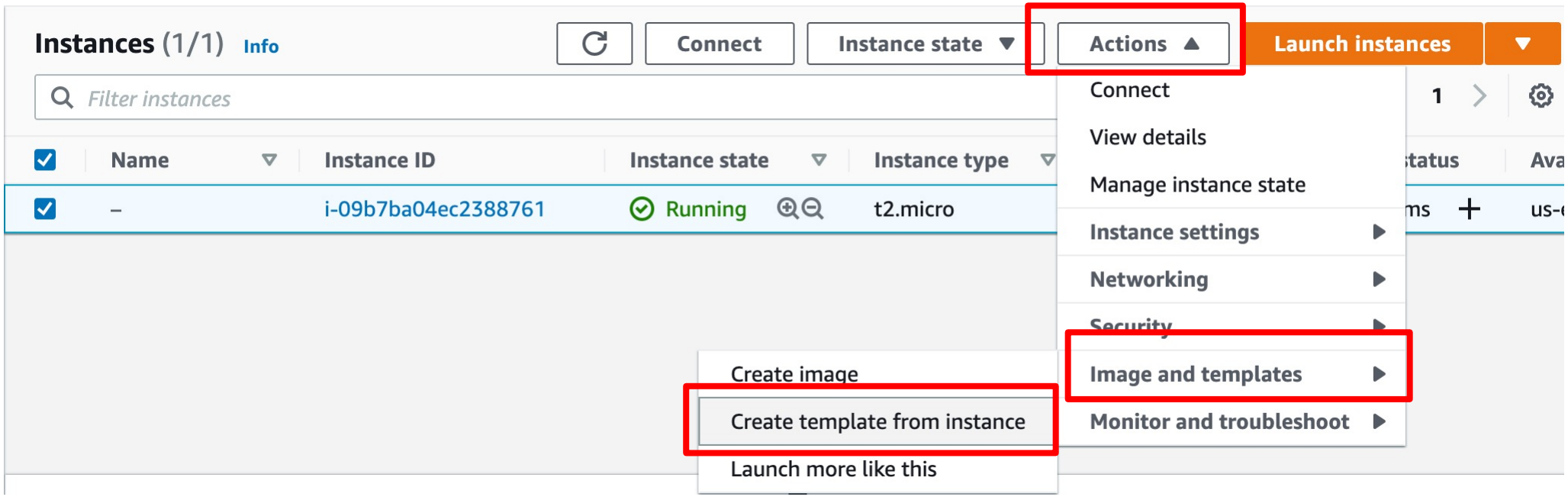
- Access to Amazon EC2 instance
  - Using “ssh”

```
$ ssh -i /your_path/amazonkey.cer ubuntu@your-public-dns
```

```
llgndeMacBook-Pro:.ssh llgn$ ssh -i amazonkey.cer ubuntu@ec2-54-175-172-23.compute-1.amazonaws.com
The authenticity of host 'ec2-54-175-172-23.compute-1.amazonaws.com (54.175.172.23)' can't be established.
ECDSA key fingerprint is SHA256:oBRUa7eguLWPkLl4JHINQGdnROk8UcqQrLlnIQo01Ec.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-175-172-23.compute-1.amazonaws.com,54.175.172.23' (ECDSA) to the list o
f known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1050-aws x86_64)

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

  System information as of Fri Sep 24 03:43:27 UTC 2021

  System load:  0.53               Processes:           101
  Usage of /:   69.3% of 9.63GB    Users logged in:     0
  Memory usage: 40%                IP address for eth0:    172.31.87.243
  Swap usage:   0%                 IP address for docker0: 172.17.0.1
```

Launch Amazon EC2 instance (Cont.)

- Create other two VMs
  - Choose the VM launched just now
  - Select “Actions” ➔ “Image and templates” ➔ “Create template from instance”



Launch Amazon EC2 instance (Cont.)

- Create other two VMs
  - Input launch template name, e.g., vmi-image-template
  - Click “Create launch template” in the right bar

Source instance

i-09b7ba04ec2388761

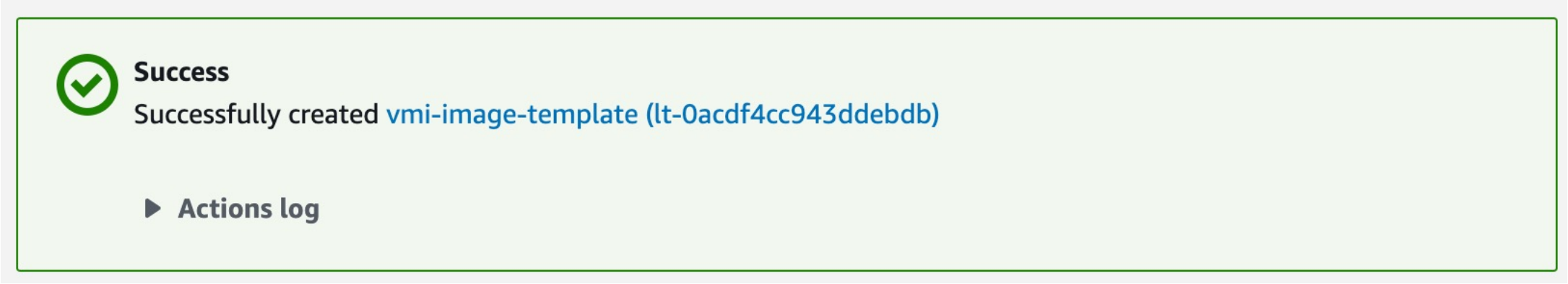
Launch template name - *required*

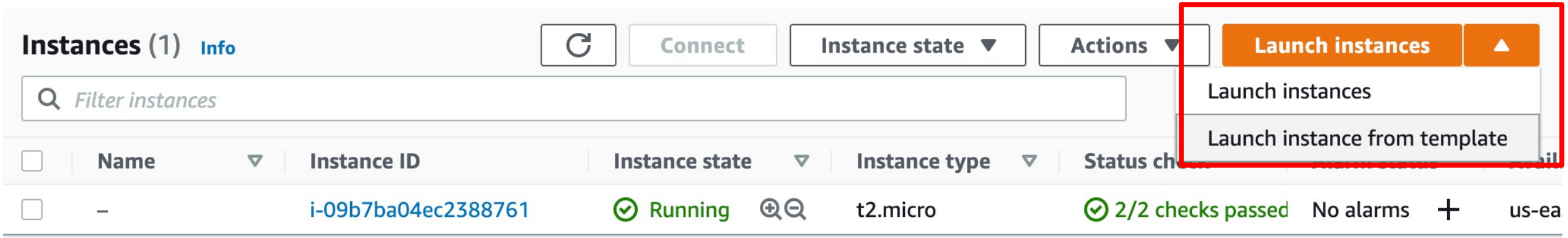
vmi-image-template

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Launch Amazon EC2 instance (Cont.)

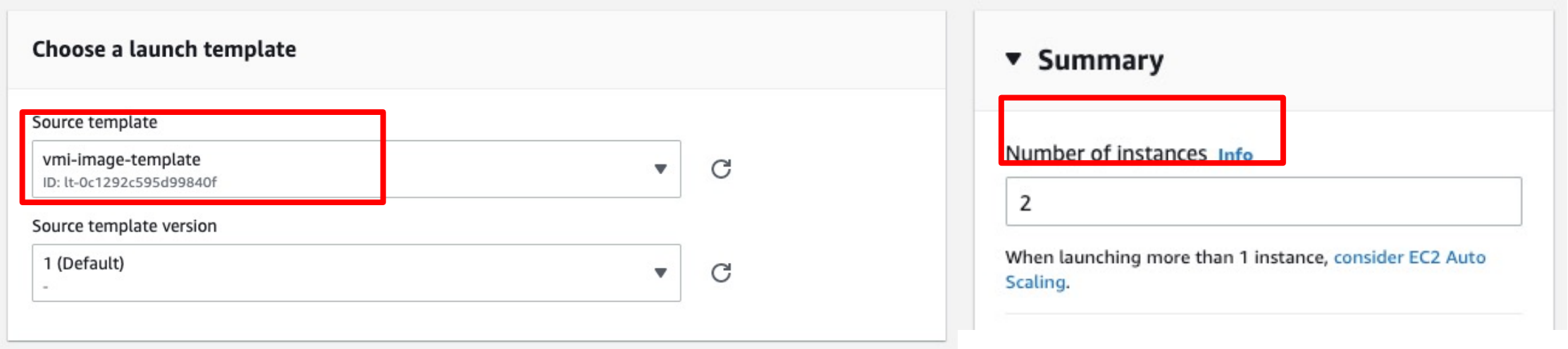
- Create other two VMs
  - After success, click “launch instances” ➔ “Launch instance from template”





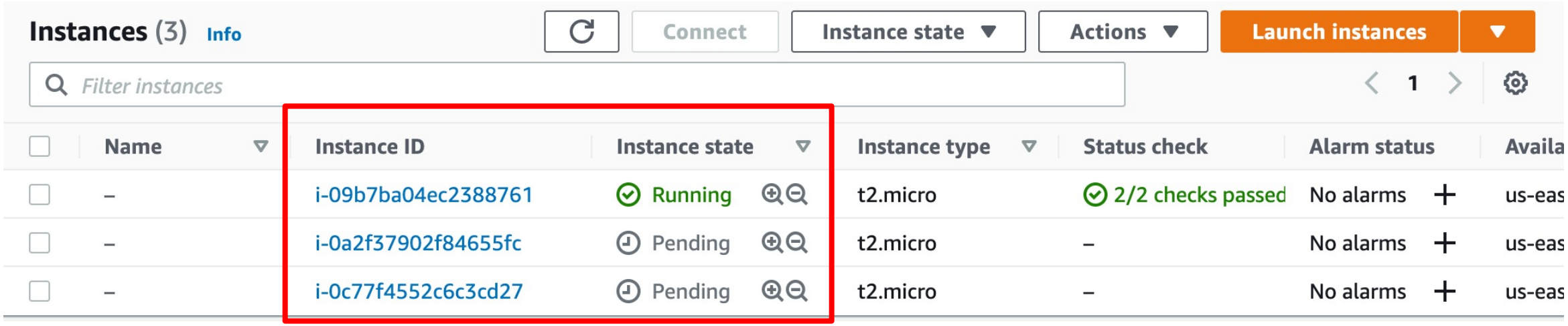
Launch Amazon EC2 instance (Cont.)

- Create other two VMs
  - Choose the template we just created and set # of instances as 2
  - Click “Launch instance”



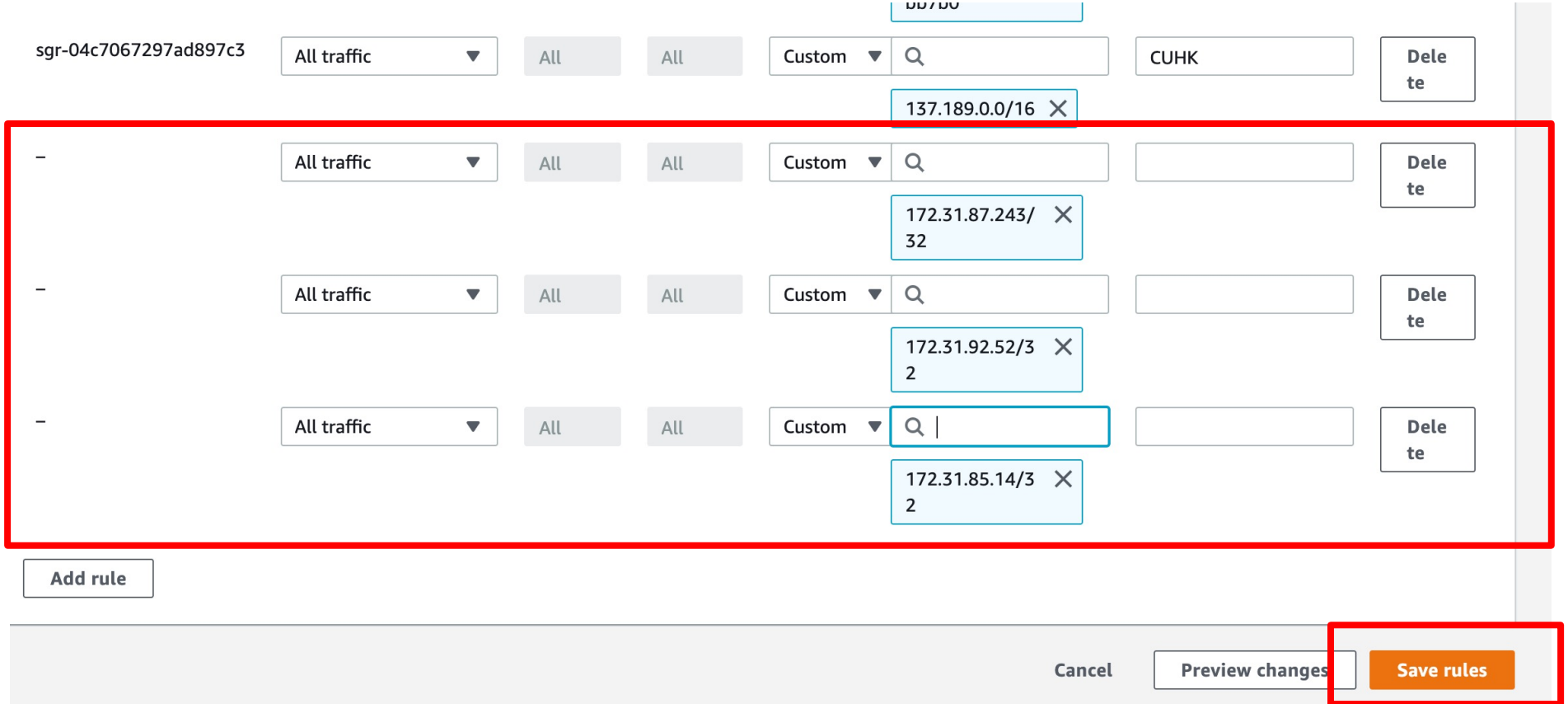
Launch Amazon EC2 instance (Cont.)

- Create other two VMs
  - View two more VMs created from template



Launch Amazon EC2 instance (Cont.)

- Configure security group
  - Enter the edit interface as mentioned before (Slide 33 and 34)
  - Add inbound rules for **the private IPs of all the three VMs** ➔ Click “Save rules”



Launch Amazon EC2 instance (Cont.)

- Configure security group
  - You can see the following rules and three VMs in the management console

Inbound rules (5)

Manage tags | Edit inbound rules

Filter security group rules

1

Name	Security group rule...	IP version	Type	Protocol
–	sgr-0b86992d6d93e8...	–	All traffic	All
–	sgr-007c03a81ea5204...	IPv4	All traffic	All
–	sgr-096f01a23c11736ea	IPv4	All traffic	All
–	sgr-04c7067297ad897...	IPv4	All traffic	All
–	sgr-02537c544f66b805d	IPv4	All traffic	All

Public IPv4 DNS	Public IPv4 address
ec2-54-175-172-23.compute-1.amazonaws.com	54.175.172.23
ec2-3-85-43-176.compute-1.amazonaws.com	3.85.43.176
ec2-3-86-111-63.compute-1.amazonaws.com	3.86.111.63

Outline

- Apply Amazon Web Services (AWS) Educate Grant
- Launch Amazon EC2 instance
- Configure Hadoop cluster on Amazon EC2
- Run MapReduce on Amazon EC2

Configure Hadoop cluster on Amazon EC2

- Access EC2 instance as mentioned before (Slide 36)
- Create new user “hadoop” (for **each** VM)
  - `$ sudo adduser hadoop`
  - Note that user “ubuntu” can sudo without password by default

```
:~ $ sudo adduser hadoop
Adding user `hadoop' ...
Adding new group `hadoop' (1001) ...
Adding new user `hadoop' (1001) with group `hadoop' ...
The home directory `/home/hadoop' already exists.  Not copying from `/etc/skel'.
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
```

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Add “hadoop” to sudo users (for **each** VM)
  - `$ sudo chmod 640 /etc/sudoers`
  - `$ sudo vim /etc/sudoers`
  - Edit and add:

    `hadoop ALL=(ALL:ALL) ALL`

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

hadoop ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
                                                          12,0-1        75%
```

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Configure ssh to log in with password (for **each** VM)
  - `$ sudo vim /etc/ssh/sshd_config`
  - Edit and change:

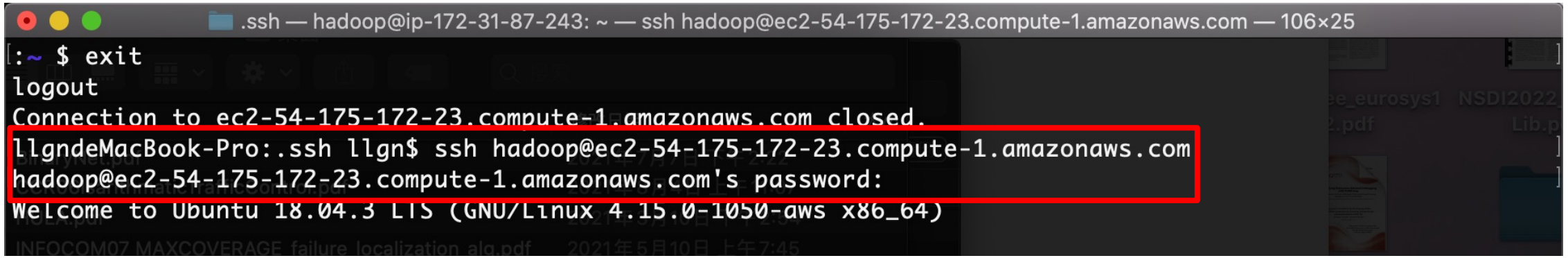
    `PasswordAuthentication yes`

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
"/etc/ssh/sshd_config" 122L, 3263C written                    56,26         42%
```

Configure Hadoop cluster on Amazon EC2 (Cont.)

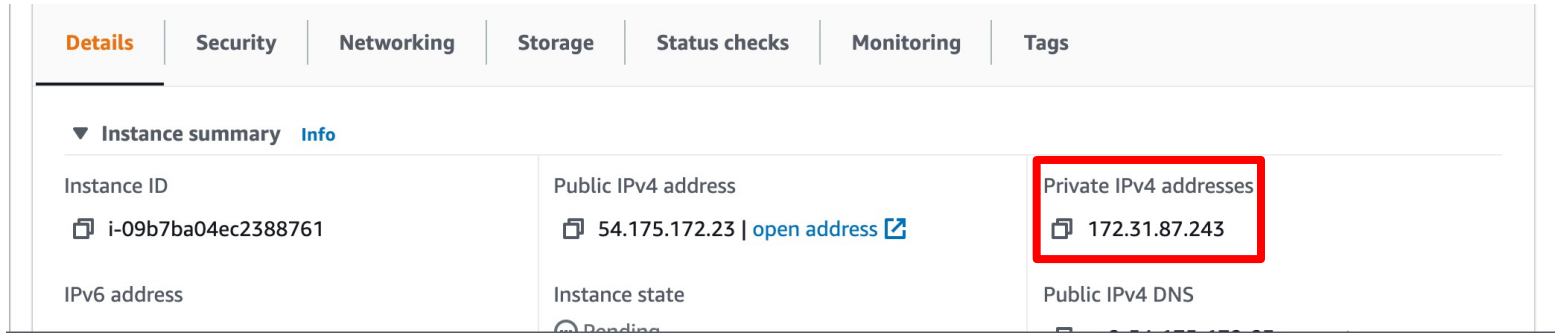
- Restart ssh service (for **each** VM)
  - `$ sudo service ssh restart`
- Log in with “hadoop” (for **each** VM)
  - `$ exit`

    `$ ssh hadoop@my-public-dsn`



Configure Hadoop cluster on Amazon EC2 (Cont.)

- Be careful to the IP address you use
  - What we can see from EC2 console:



  - What we can see by running `ifconfig`:

```
hadoop@ip-172-31-87-243:~$ ifconfig -a
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:be:1f:bd:38  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.87.243  netmask 255.255.240.0  broadcast 172.31.95.255
        inet6 fe80::1057:26ff:fe71:1bb5  prefixlen 64  scopeid 0x20<link>
        ether 12:57:26:71:1b:b5  txqueuelen 1000  (Ethernet)
```

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Be careful to the IP address you use.
  - Public IP is not bound to the IP stack of instance's OS.
    - So we should not use public IP, but use **Public DNS**, or **Private IP** instead when configuring Hadoop
    - We suggest **private IP** since it is **static**
    - Ref: Hadoop config on EC2: why public DNS works but not public IP?

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Configure /etc/hostname for **each** VM
  - Please set hostname as "**vm1**", "**vm2**", "**vm3**" respectively
  - Edit hostname for **each** VM, e.g., on **vm1**
    - `$ sudo vim /etc/hostname`
    - `$ sudo hostname vm1`
    - Log out and re-log in your vm

```
hadoop@ip-172-31-87-243:~$ sudo hostname vm1
hadoop@ip-172-31-87-243:~$ exit
logout
Connection to ec2-54-175-172-23.compute-1.amazonaws.com closed.
llgndeMacBook-Pro:Desktop llgn$ ssh hadoop@ec2-54-175-172-23.compute-1.amazonaws.com
hadoop@ec2-54-175-172-23.compute-1.amazonaws.com's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1050-aws x86_64)
```

```
Last login: Fri Sep 24 04:09:02 2021 from 137.189.241.71
hadoop@vm1:~$
```

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Configure /etc/hosts for **each** VM
  - `$ sudo vim /etc/hosts`
  - Write down the mapping between the **private IPs** and hostname

    eg:

```
172.31.87.243 vm1
172.31.92.52 vm2
172.31.85.14 vm3
```

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Pre-requisite (for **each** VM, detailed in previous tutorials)
  - Java
    - Download jdk-8u144-linux-x86.tar.gz in your local machine from https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html
    - `$ scp jdk-8u144-linux-x64.tar.gz hadoop@your_public_dns:/home/hadoop`
    - `$ tar -zxvf /home/hadoop/jdk-8u144-linux-x64.tar.gz`
    - Add "export JAVA_HOME=/home/hadoop/jdk1.8.0_144" and "export `PATH=$JAVA_HOME/bin:$PATH`" to "~/.bashrc" in VM
    - `$ source ~/.bashrc`
    - `$ java -version`

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Pre-requisite (for **each** VM, detailed in previous tutorials)
  - Hadoop
    - Download Hadoop 2.7.3 in your local machine from https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz
    - `$ scp hadoop-2.7.3.tar.gz hadoop@your_public_dns:/home/hadoop`
    - `$ tar -zxvf /home/hadoop/ hadoop-2.7.3.tar.gz`
    - Add
      - "export HADOOP_HOME=/home/hadoop/hadoop-2.7.3"
      - "export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar"
      - "export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH"

      to "~/.bashrc" in VM
    - `$ source ~/.bashrc`
    - `$ hadoop version`
  - Code of WordCount
    - You can also use scp to transmit from your local machine to the remote VM

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Configure password-free ssh connection (for **each** VM)
  - Take vm1 as an example
    - `$ ssh-keygen`
    - `$ ssh-copy-id hadoop@vm2`
    - `$ ssh-copy-id hadoop@vm3`
    - `$ ssh hadoop@vm2` for test
    - `$ ssh hadoop@vm3` for test
  - If you encounter an error like “The ECDSA host key vm3 has changed”
    - It is due to the cache of known hosts (the mapping between public key and hostname)
    - `$ ssh-keygen –R vm3`

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Set up Hadoop cluster (for **each** VM)
  - Update configure files under of Hadoop (see details in Tutorial 2)

```
$ scp /local_path/csci4180_tuto2.tar.g hadoop@your_public_dns:/home/hadoop
$ tar -xzvf /home/hadoop/csci4180_tuto2.tar.gz
$ vim csci4180_tuto2/csci4180_hadoop_conf/hadoop-env.sh
(Change "export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_144" to
"export JAVA_HOME=/your_path/jdk1.8.0_144")
$ cp csci4180_tuto2/csci4180_hadoop_conf/* hadoop-2.7.3/etc/hadoop/
```

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Check if Hadoop runs properly (**for vm1 only**)
  - Init Hadoop
    - `$ hadoop namenode -format`
  - Start Hadoop
    - `$ start-dfs.sh`
    - `$ start-yarn.sh`
  - Test Hadoop
    - `$ hdfs dfsadmin -report`
  - (The same as tutorial 2)

Configure Hadoop cluster on Amazon EC2 (Cont.)

- Reset Hadoop
  - `stop-dfs.sh, stop-yarn.sh` (**only in vm1**)
  - Clean the hadoop folder in **vm2 and vm3** (i.e., datanodes);
    - rm -rf /your_path/hadoop-2.7.3/tmp
  - Re-copy the hadoop folder from vm1
  - Restart Hadoop (**only in vm1**)
  - `$ hadoop namenode -format`
  - `$ start-dfs.sh`
  - `$ start-yarn.sh`

Outline

- Apply Amazon Web Services (AWS) Educate Grant
- Launch Amazon EC2 instance
- Configure Hadoop cluster on Amazon EC2
- Run MapReduce on Amazon EC2

Run MapReduce on Amazon EC2

- Run MapReduce (See details in Tutorial 2)
  - WordCount example
    - Assuming that
      - **/user/hadoop/wordcount/input** - input directory in HDFS, which contains the input file for MapReduce
      - **/user/hadoop/wordcount/output** - output directory in HDFS, make sure it does NOT exist before running MapReduce job
  - Run **in vm1**

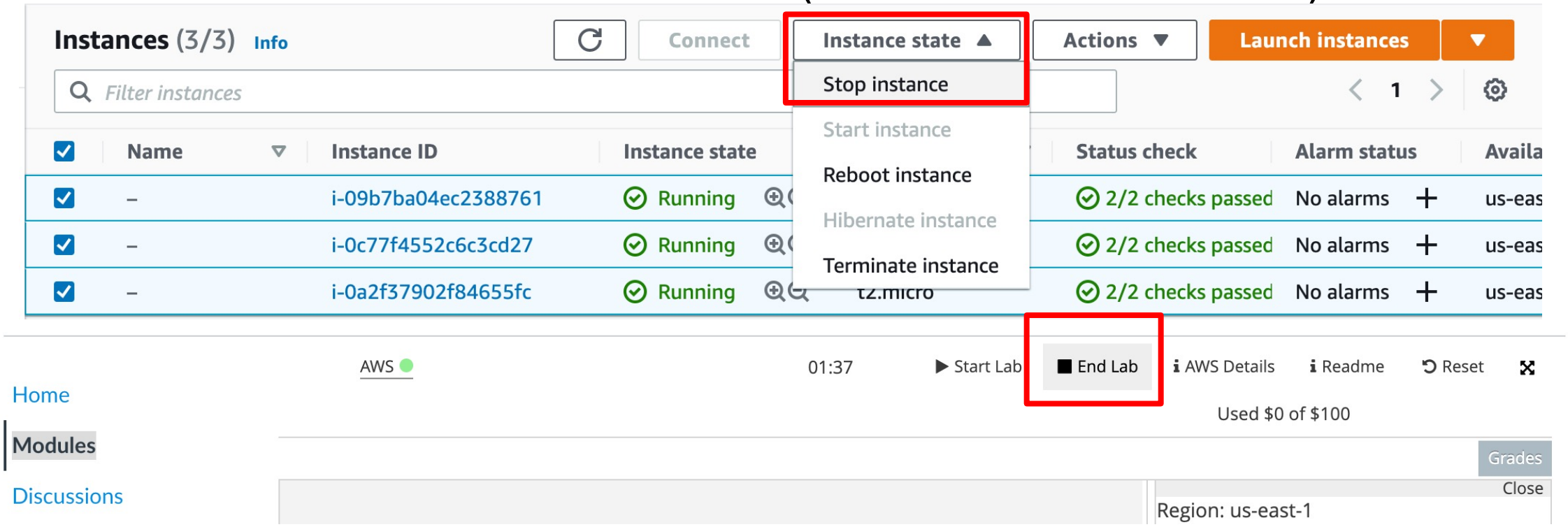
```
$ mkdir ~/wordcount
$ cp ~/csci4180_tuto2/WordCount.java ~/wordcount
$ cd ~/wordcount
$ hadoop com.sun.tools.javac.Main WordCount.java
$ jar cf wordcount.jar WordCount*.class
$ hadoop fs -mkdir -p /user/hadoop/wordcount
$ hadoop fs -put input /user/hadoop/wordcount/input
$ hadoop jar wordcount.jar WordCount /user/hadoop/wordcount/input
/user/hadoop/wordcount/output
```

Run MapReduce on Amazon EC2 (Cont.)

- Problems when running WordCount on VM (Very slow)
  - You may encounter the problem (e.g. pending or very slow) due to **lack of memory**
    - Each VM only has 1GB RAM
  - You can use the following command line to clean memory cache under **root** user.
    - `$ sudo -i`
    - `$ echo 3 > /proc/sys/vm/drop_caches`
  - Another solution is to kill the memory-consuming tasks
    - You can use htop to observe CPU-/memory-/disk-consuming tasks
    - For example, if there are many threads running “mysqld” in the background
      - `$ sudo service mysql stop`

Run MapReduce on Amazon EC2 (Cont.)

- After using the instance, **stop** it to avoid extra charge
  - Note: "Terminate" indicates to **delete** your VM.
    - **Data will be lost after terminating it**
  - Also click "End Lab" in the terminal (as mentioned in Slide 15)



Summary

- Setup the Amazon Web Services (AWS) Canvas account
  - Get the AWS Educate Grant with the invitation link
- Launch Amazon EC2 instance
  - Configure the instance
  - Three Ubuntu-18.04 instances
- Configure Hadoop cluster on Amazon EC2
  - Similar to Tutorial 2
  - Be careful to IP addresses
- Run MapReduce on Amazon EC2
  - Similar to Tutorial 2