Project 1.a Create an AWS Instance, setup firewall restriction of server access

For this specialization, I have created an Amazon Machine Image (AMI), coursera_cs5910_im2 with Linux-Apache-MySQL-PHP (LAMP) server package and GnuPG 2.4.0 installed for you to clone an instance and use it for learning the cybersecurity concepts, security policy and related enforcement procedures. You can clone an instance with coursera_cs5910_im2 in about 2 minutes.

You will need an AWS account to login to the AWS management console and use the EC2 GUI control to create the instance for our class projects.

Your first task is to apply for your free AWS Academy account. Follow the instruction in Section 1. The AWS academy account allows you to work on the creation of AWS EC2 instances for the exercises of this specialization. However, it restricts the usage of AWS service only in us-east-1 (North Virginia) region. Note that the coursera_cs5910_im2 image is only available on N. Virginia region.

Warning: Note that with your free AWS Academy account, you need to use your $100 free credit wisely. Make sure to stop your instances after your session and stop your lab asap. Developing good public cloud hygiene habit is important.

In this project, you will learn how to set up a default project web page. You will learn how to **restrict access to services** of the instance only to you at home by specifying the sources IP address of the firewall rules using the Security Group Interface of the instance. You will also learn how use SSH command with the private key to access the AWS Linux instance without providing login or password.

1. Create AWS Account

1.1 Create free AWS Academy Account with Basic Support.

For this specialization and certificate, I have worked with AWS Academy to provide an AWS Academy Student Account with $100 free credit for you to conduct the exercises in this specialization. You will not need to show your credit card for the accessing this AWS Academy account. I will need you to email cchow@uccs.edu with subject titled "request AWS Academy account for xxxxxx Coursera course" and include the browser image or the official email from Coursera which proves your enrollment of our Coursera courses. Make sure you indicate clearly the registered email address, which I should use as Username for you to access the AWS Academy portal.

When I added you to the AWS Academy Learner Lab, you will receive an email similar to the one below:

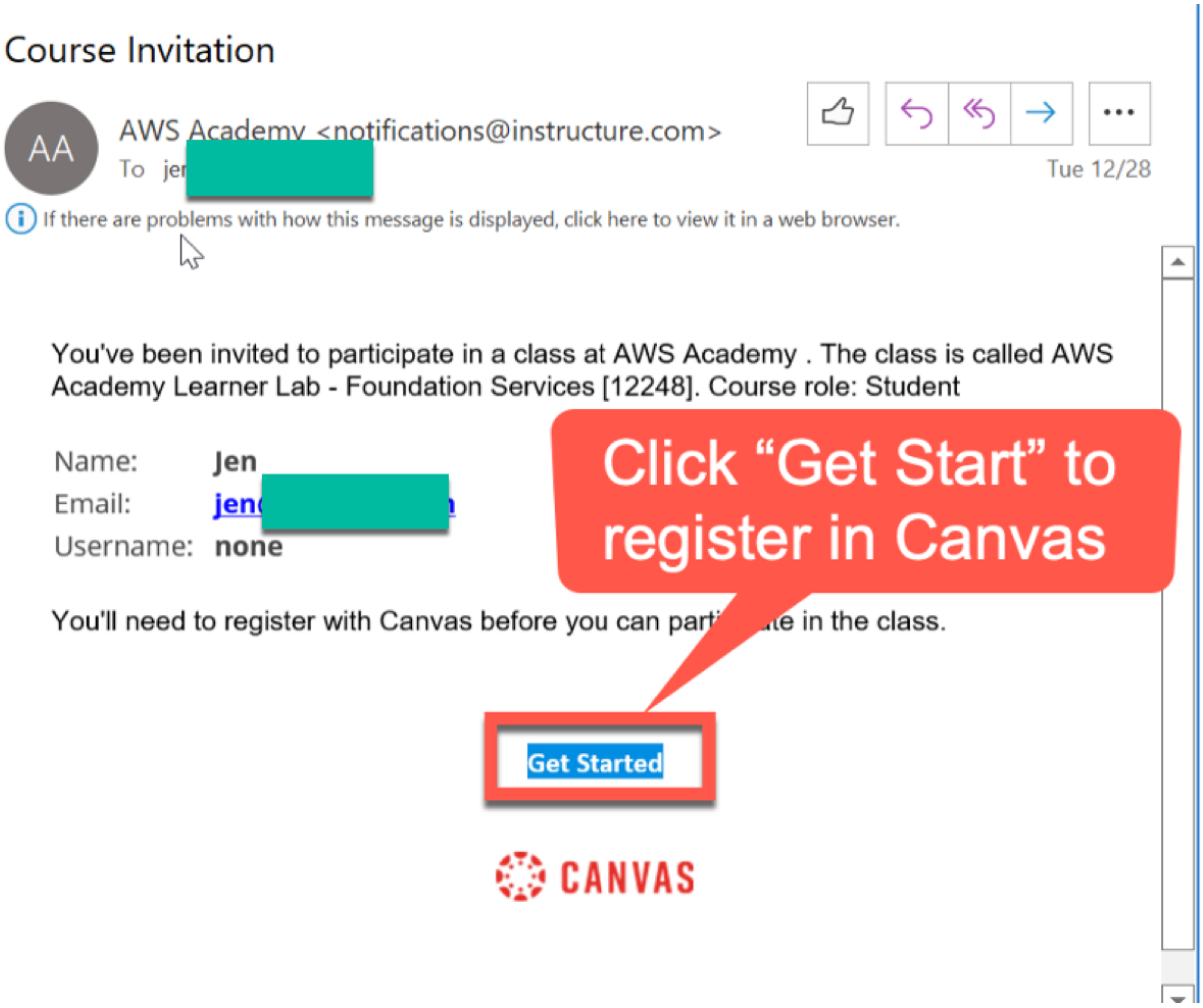

Figure 1. Get Started to register in Canvas Learning Management System.

Click the “Get Started” button. You will be directed to a web site with url similar to

https://awsacademy.instructure.com/courses/12xxx8?invitation=tuuJ6s3gcaYt92sxycafaXXX

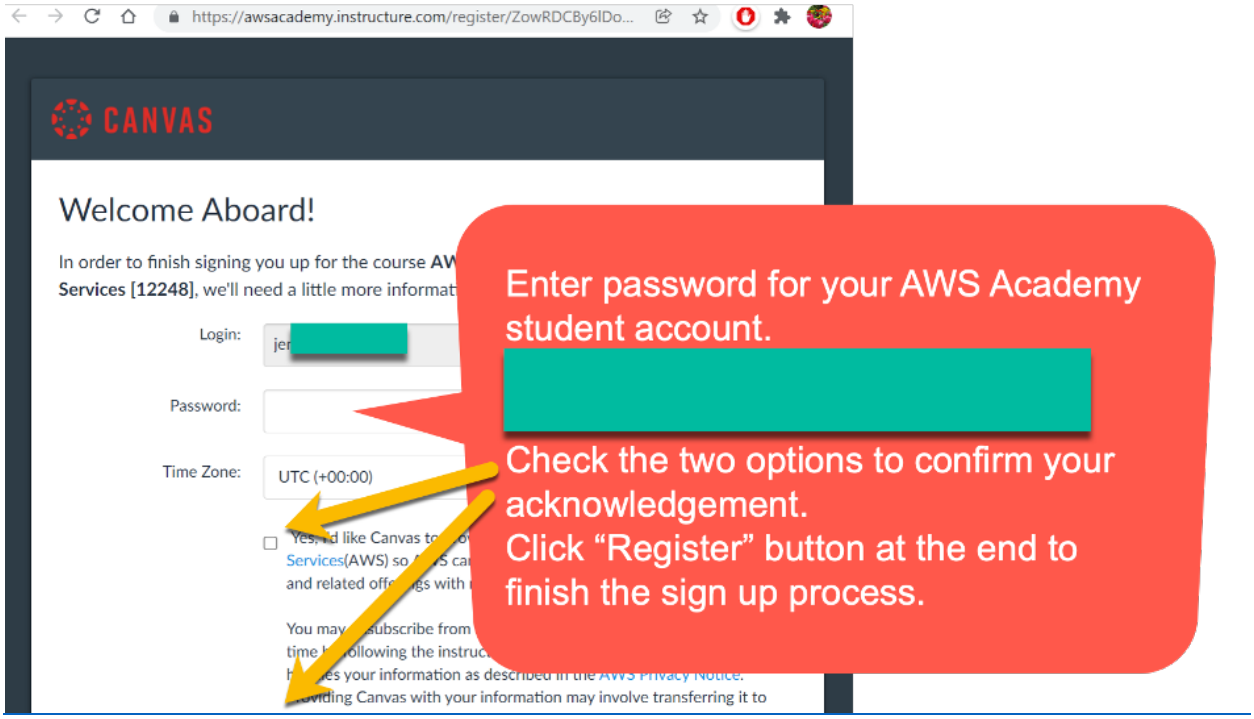

Figure 2. Specify your password for the AWS Academy account.

Enter the password for your AWS Academy student account. Check the two options to confirm your acknowledgement. Click "Register" button at the end to finish the sign-up process.

Follow the above steps to register with Canvas, which is a Learning Management System (LMS) for AWS Academy.

Note that with your AWS Academy account, you will not be asked for credit card and set up a regular AWS account with billing. Your access will go through AWS Academy web site and then through vocareum.com 3rd party web site to access free AWS services. Please do not go directly to https://aws.com/ then click the management console on the right side. In that case you will be asked to setup regular account with your credit card.

Here are the six steps where you can access the AWS Management Console with free credits for creating AWS EC2 instance to be used for our class projects:

1. Login to your AWS Academy account. https://www.awsacademy.com/SiteLogin
2. Click "Modules"

3. Click “Learner Lab – Foundational Services”

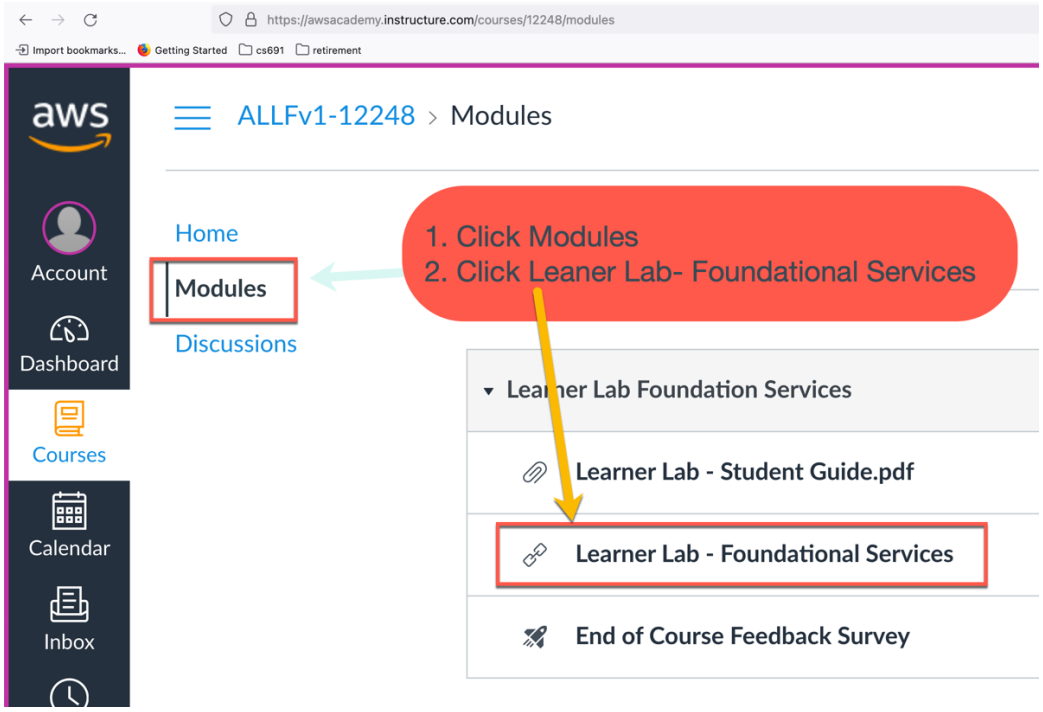

Figure 3. Select Learner Lab – Foundational Services Module

4. Click “Start Lab”. Watch the circle status icon to the right of AWS change from red to yellow to green (ready). Click AWS when its right icon turns green. It will take 2-3 minutes to setup a working session on AWS through the vocareum.com interface.

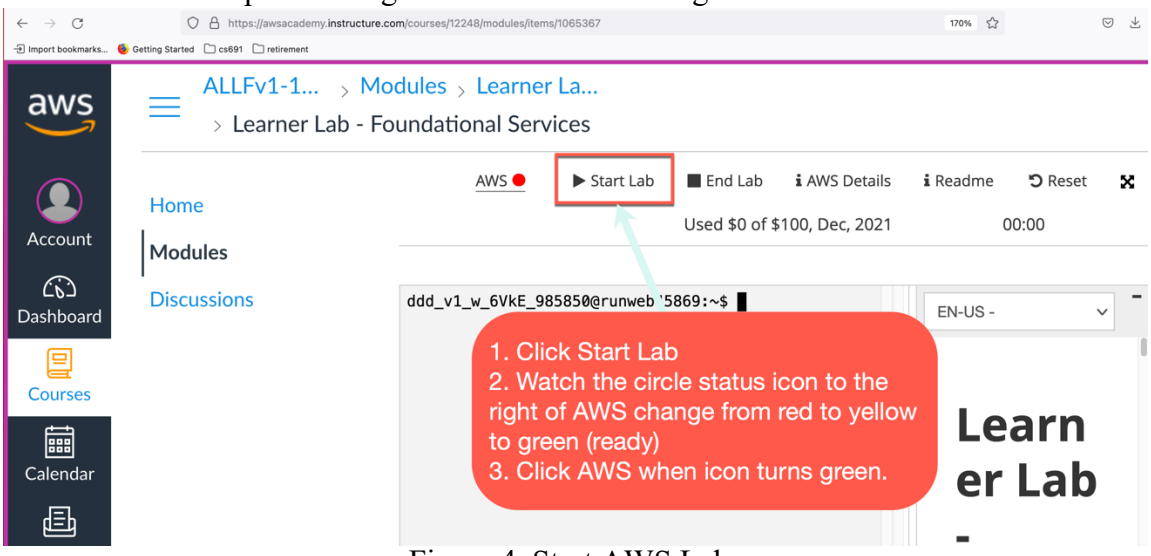

Figure 4. Start AWS Lab.

5. Click AWS to start AWS management control for using the ec2 service and creating instance. Watch for used credits and the session countdown time.

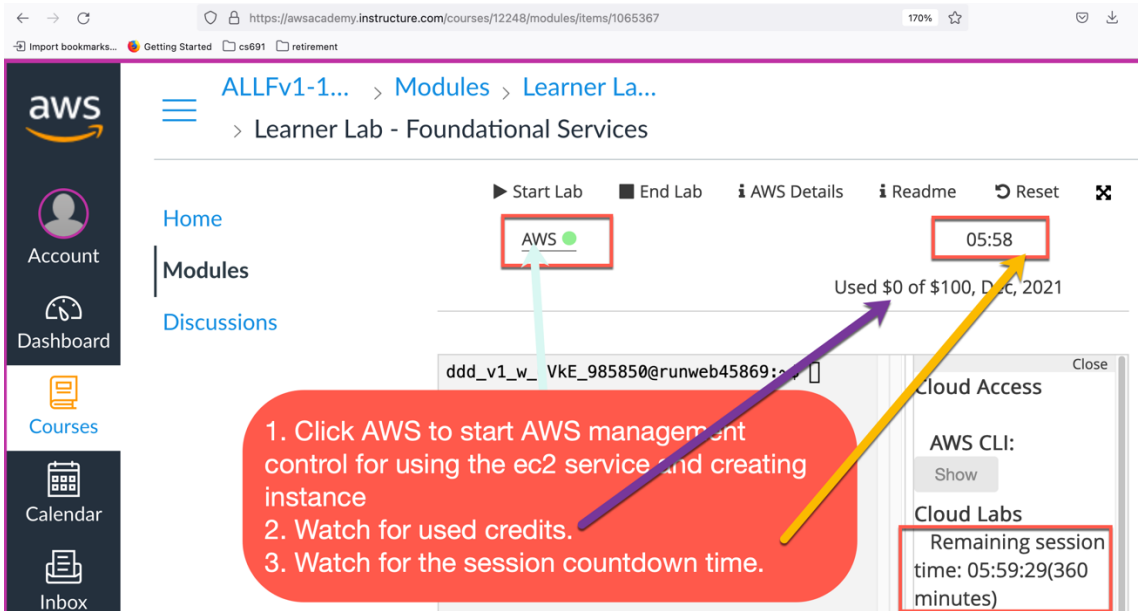

Figure 5. Click AWS when the dot turns green. Watch statistics of your account.

6. You are now accessing AWS services through an account under vocstartsoft with an ID and your email. It also shows that you are accessing the default and only region, N. Virginia. Select EC2 AWS service.

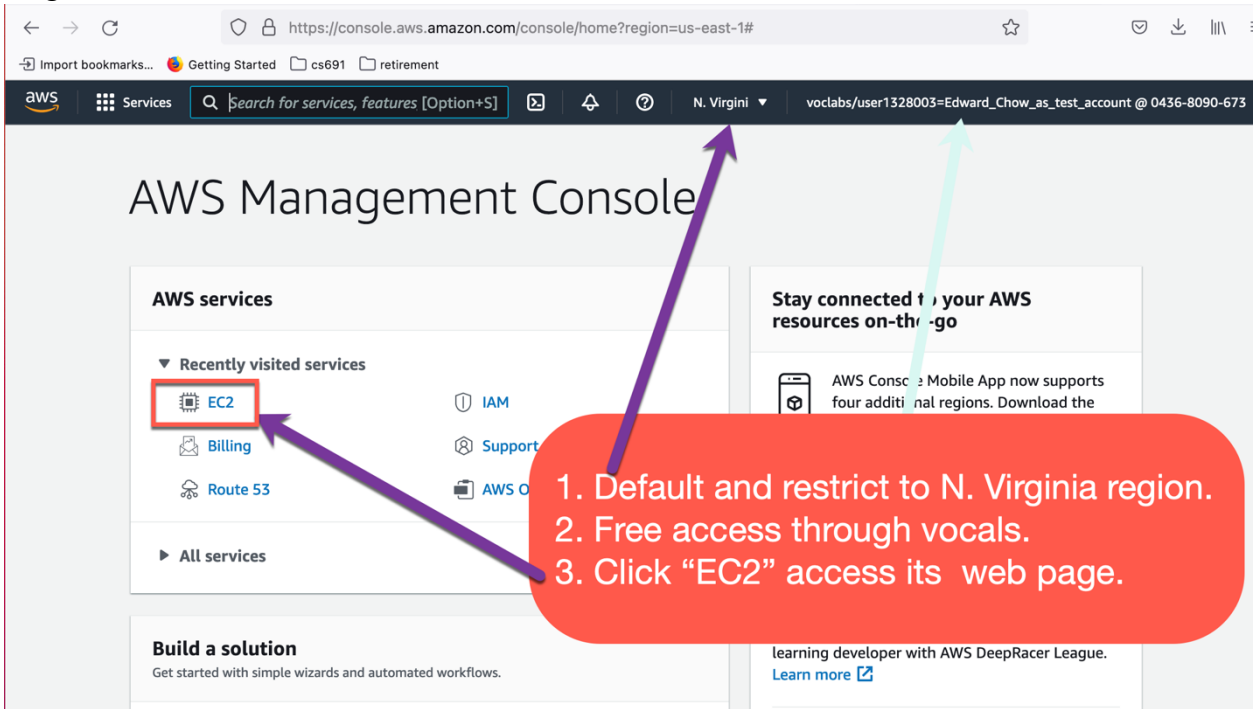

Figure 6. Click EC2 to access its dashboard interface.

2. Create AMI instance and Install LAMP for CS5910 exercises.

2.1 Create an Amazon Linux 2 instance from coursera_cs5910_im2

Once setup an AWS account, use the AWS management console
to create an AWS EC2 instance. Click on EC2 service. Note that for those with AWS Academy account (shown in the interface below as voclabs account), you region will be automatically set to N. Virginia and you will not be able to use services outsides of that region.

Then pull down the "Launch instance" menu and choose "Launch instance" menu item.

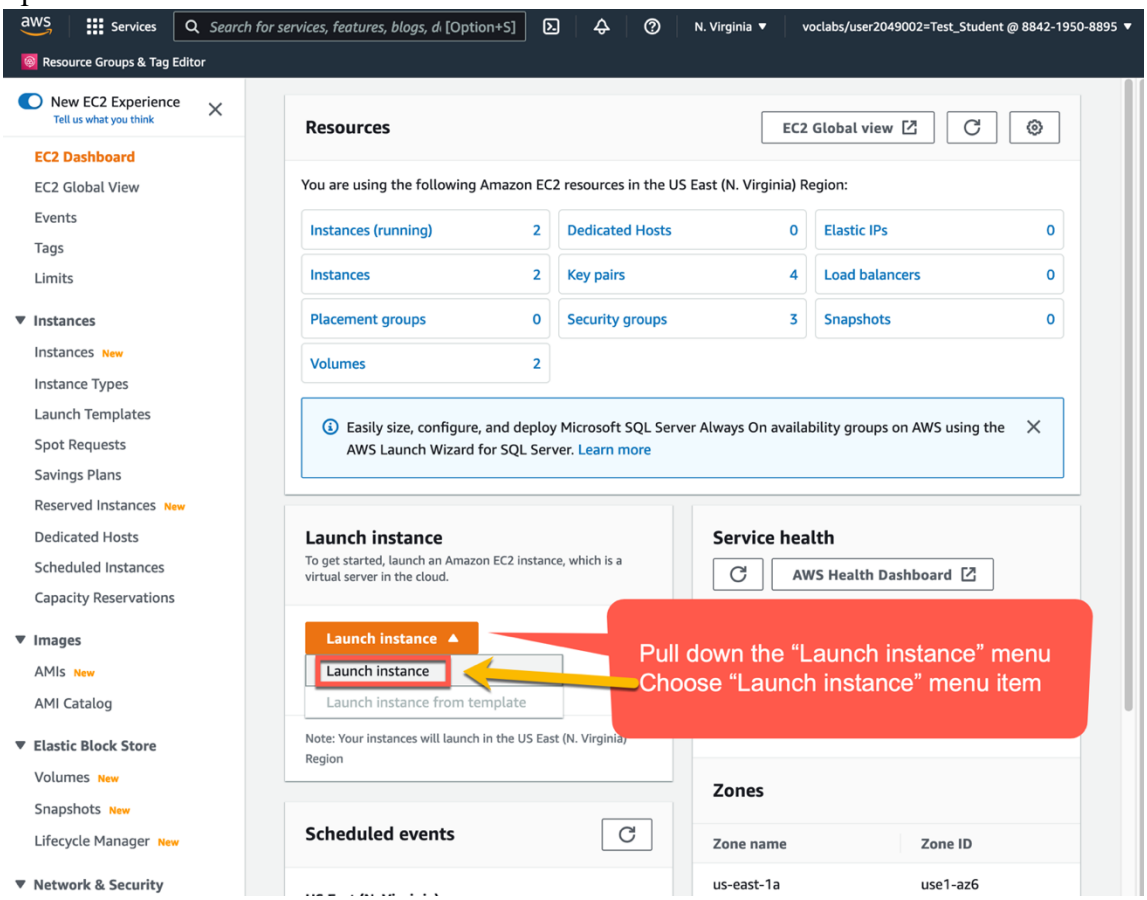

Figure 7. Select "Launch instances" menuitem.

We will then be asked to enter different parameters and specification of EC2 instance you like to clone in the next dialog window. The parameters are organized in different sections in the next interface.

1. **Name and tags section.**
   Enter "<your login>_ awsac_cs5910_i1" for the value of Name tag associated with the instance. Here replace <yourlogin> with the login part of your email address. I will enter cchow_awsac_cs5910_i1. The naming includes the class name cs5910 and i1 for instance #1. This will help you distinguish different instances for different classes on the EC2 dashboard control panel. Note that the name tag values will be associated with the instances and show up the first column in the instance list.
2. **Application and OS images Section.**
   Enter "coursera" in the query box to search for the coursera_cs5910_im2 image in a

shortlist.

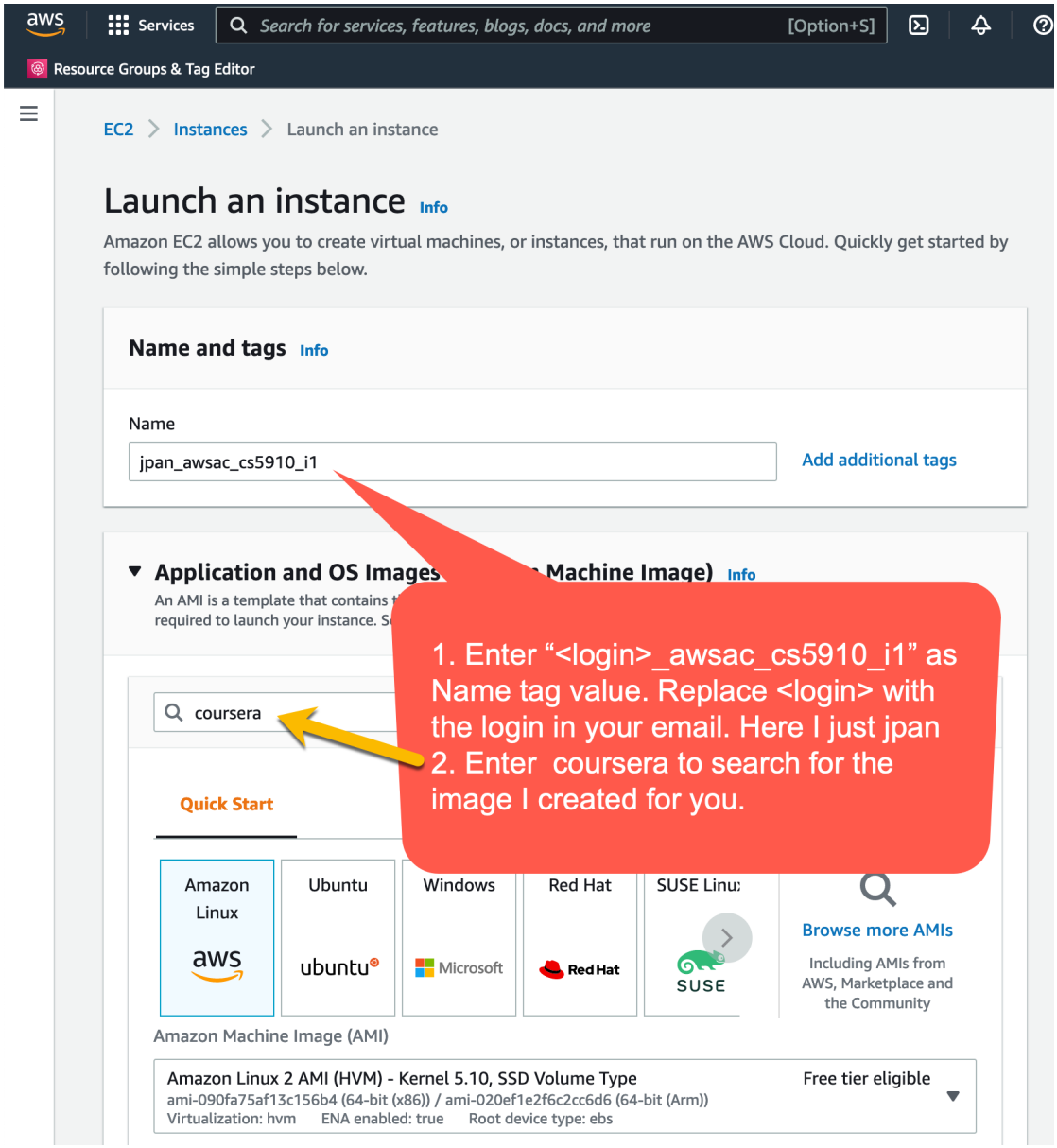

Figure 8. Specify the name of instance and search for image with prefix coursera.

The list of all public community images will show up. Select coursera_cs5910_im2. It includes LAMP servers, GnuPG 2.4.0, and software development tools.

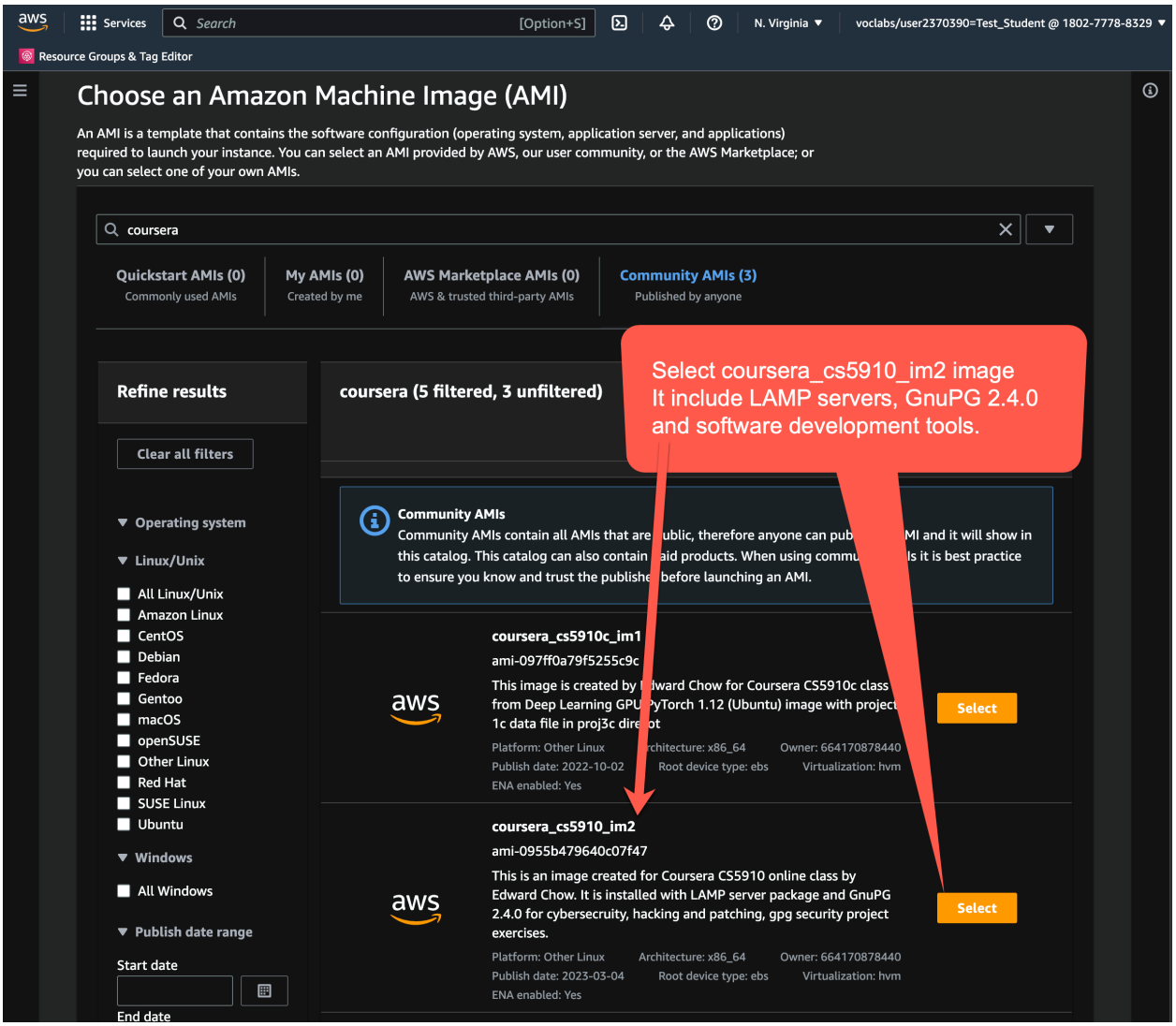

Figure 9. Choose coursera_cs5910_im2 image to clone.

Next scroll down this window and choose

3. **Instance Type Section.**
   Choose t2.micro as instance type so that we got charged less.
4. **Key pair (login) Section.**
   This deals with critical SSH secure access to the instance.
   Click “Create new key pair”.

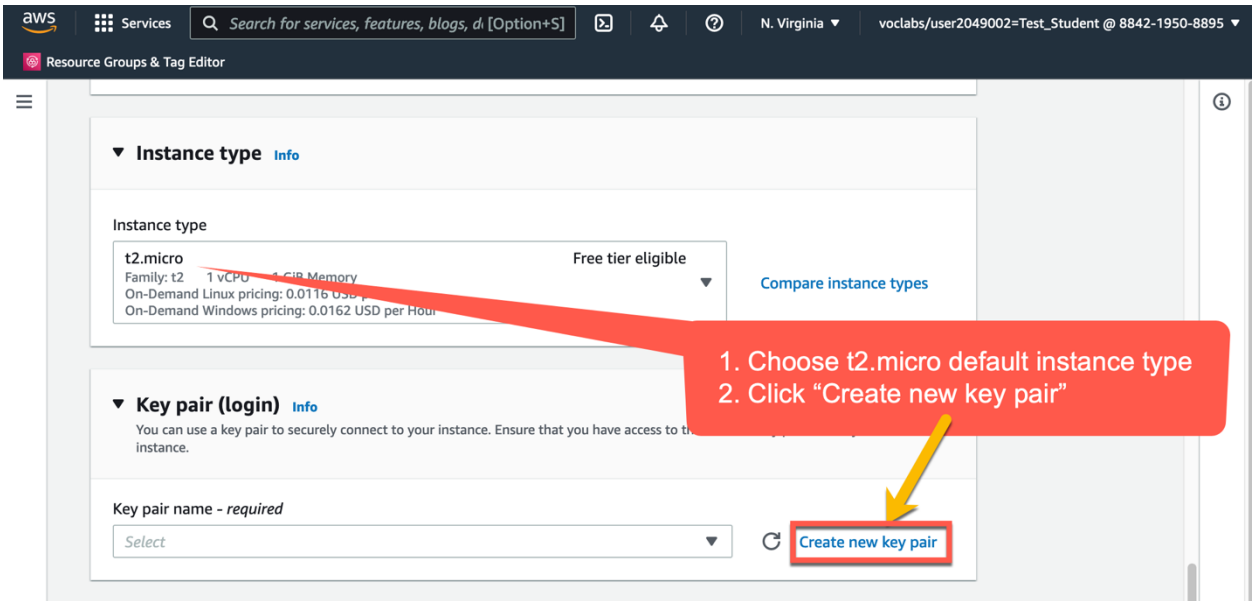

Figure 10. Choose to create new key pair.

Enter "<yourlogin>_awsac_cs5910_pkey" as Key pair name. Choose default RSA key pair type and .pem key file format if you are using OpenSSH type to connect to the instance. If you use PuTTY as terminal app to access your instance, you should pick .ppk file format for your key. I enter "jpan_awsac_cs5910_pkey" for my key pair name in the following popup window assume jpan is the login part of my email address. When click the "Create key pair" button, the filename jpan_awsac_cs5910_pkey.pem will be used to save the private key content of the public key pair.

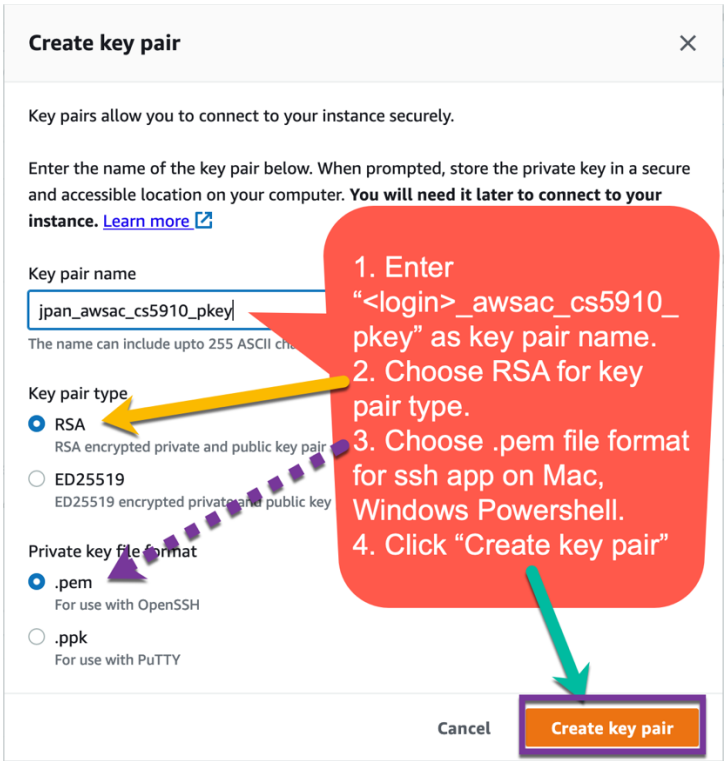

Figure 11. Specify Key pair name and the Key pair type.

The private key file of the public key pair will be automatically generated and downloaded to your local client. On mac it will be saved in the Download folde.

!!!Try to save and back it up to a safe place. Since it is not encrypted, you may want to encrypt it. This is critical. The AWS will not offer another chance to download this private key.!!!

Note that the related public key of this public key pair will be saved in the /home/ec2-user/.ssh/authorized_keys file for future verification. The above private key will be used to encrypt a security token to be sent to the instance for verification. The security token will be decrypted with the public key using the RSA algorithm as one of the inputs during the verification.

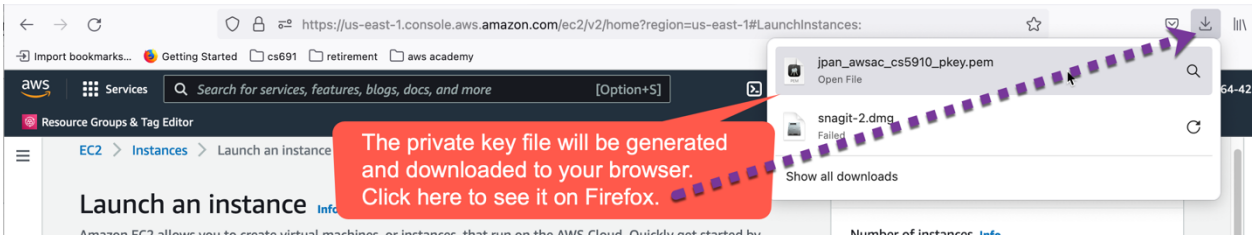

Figure 12. Downloaded private key in the local client.

5. Network settings Section.

This deals with network and firewall (security group) settings.
Here we will pick the defaults, choose to Create security group, select Allow SSH, HTTPS, and HTTP traffic by checking the boxes to their right. Then click Edit button on the upper right to restrict incoming traffic to SSH, HTTPS, and HTTP ports only from your local client. We use an option provided called MyIP. It is our local client machine public IP address. AWS EC2 will automatically set it based on the source IP address in the packets received from you. We do not want hackers to access our servers when they are not yet patched. Also the image contains vulnerable web apps for future project1c hacking and patching cybersecurity exercises. It can be easily hacked if we open the instance for wider Internet access than just your client machine.

Later on after we finish the system update and install all security patches, we can open up by adding additional subnets or IP address which are allowed to come in. It is also possible that our client IP address has been changed because we move to a different subnets (says from home to school) or our ISP somehow reassigned new public IP address for our home network, then we need to follow the steps in Section 3.3 Pages 41-43 to reset the My IP value.

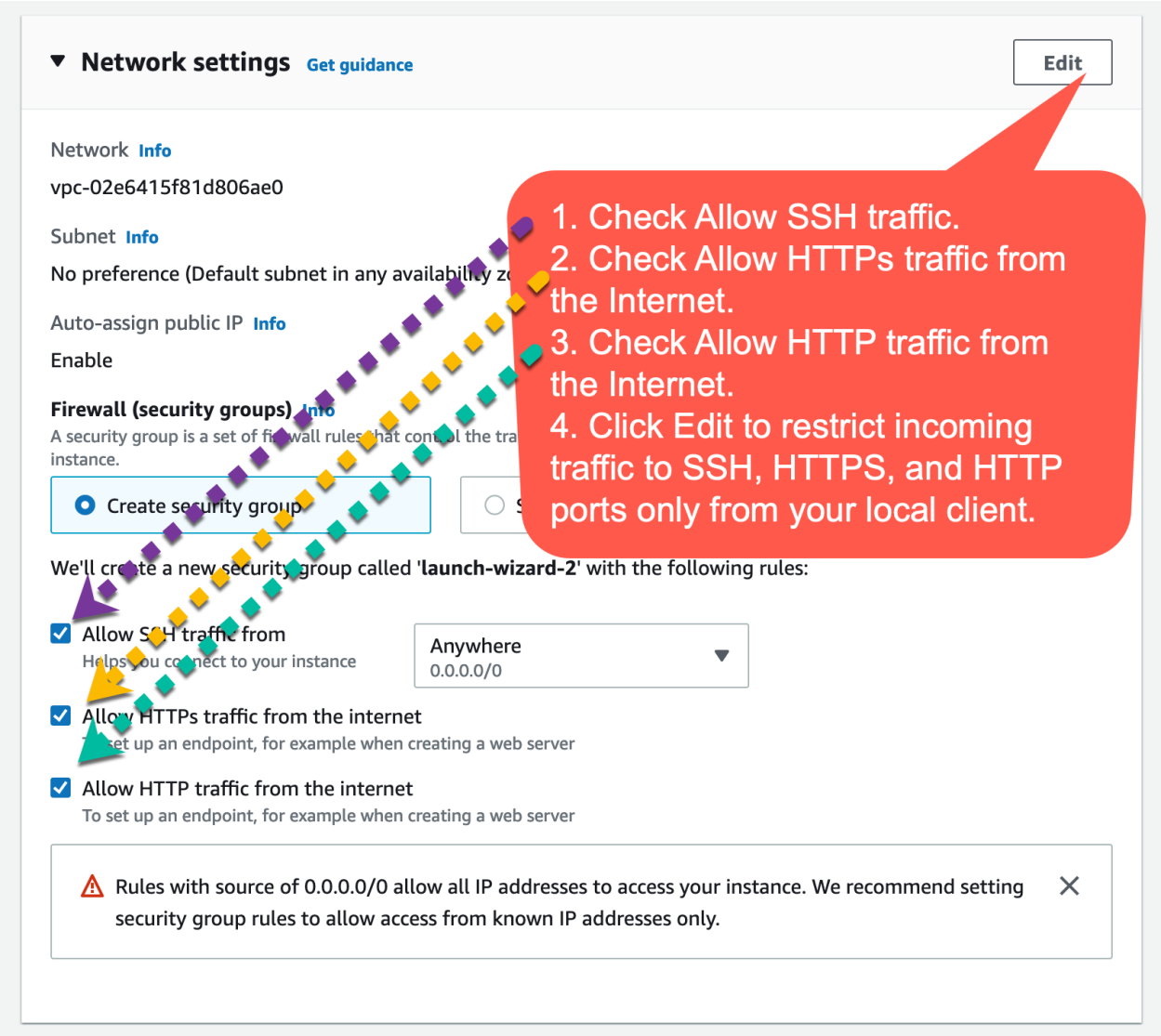

Figure 13a. Select Edit network settings parameters.

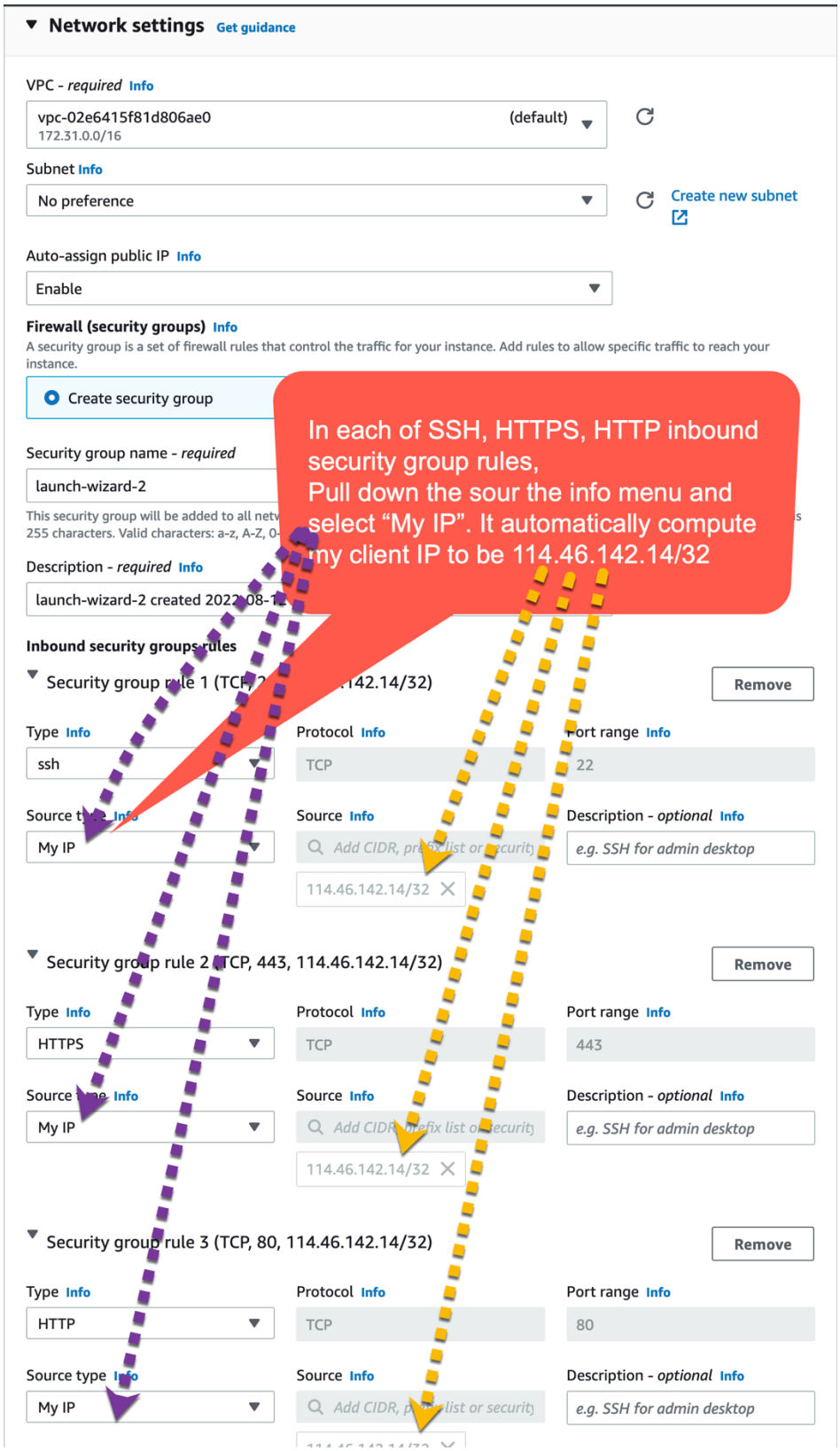

Figure 13b. Set Source type to My IP for SSH, HTTPS, and HTTP.

5. **Configure storage Section.**
   Choose the defaults. No changes.
6. Scroll down and click "**Launch instance**" button to conclude the instance setting.

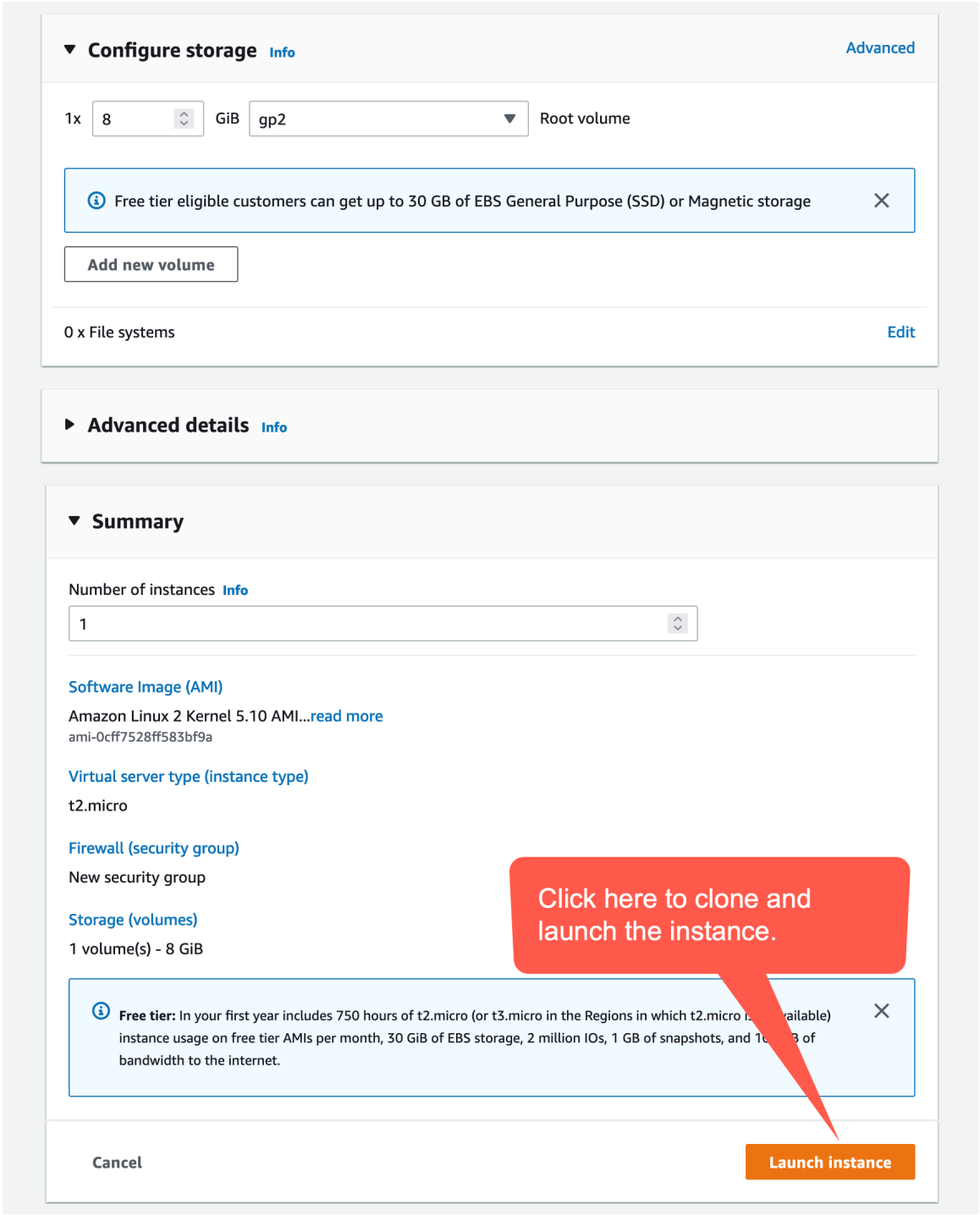

Figure 14. Click "Launch instance".

Soon you will see the web page indicating your instance is successfully initiated. We will click "View all instances" button to see the EC2 dashboard with all your instances.

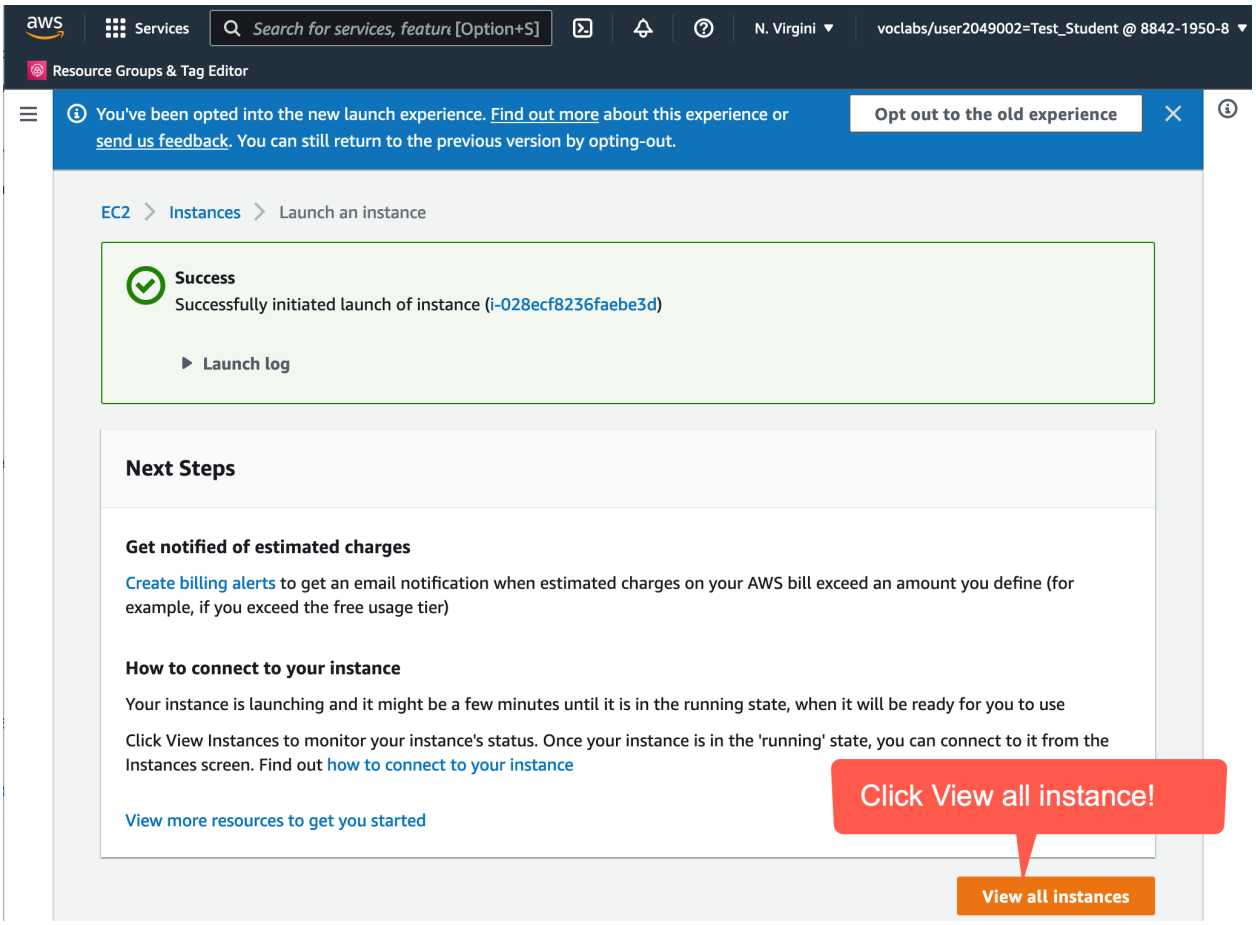

Figure 15. View instance just created.

If you can not find the instance with the name <yourlogin>_awsac_cs5910_i1, you can search for the last one in the list and verify its AMI name, Launch time, keypair name to be sure. Then you can click on the Name column to enter the instance name.

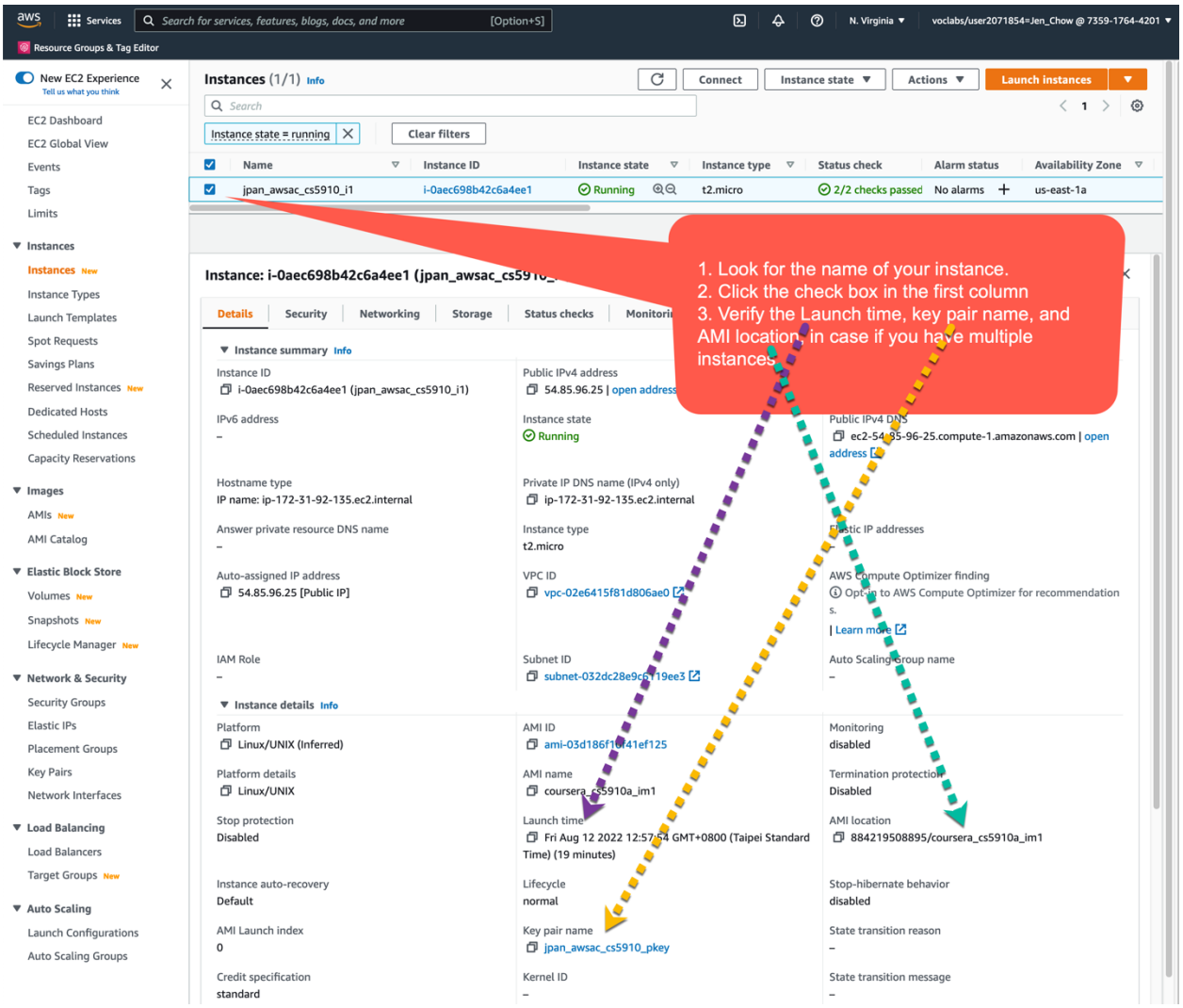

Figure 16. Find the instance and check its detailed info.

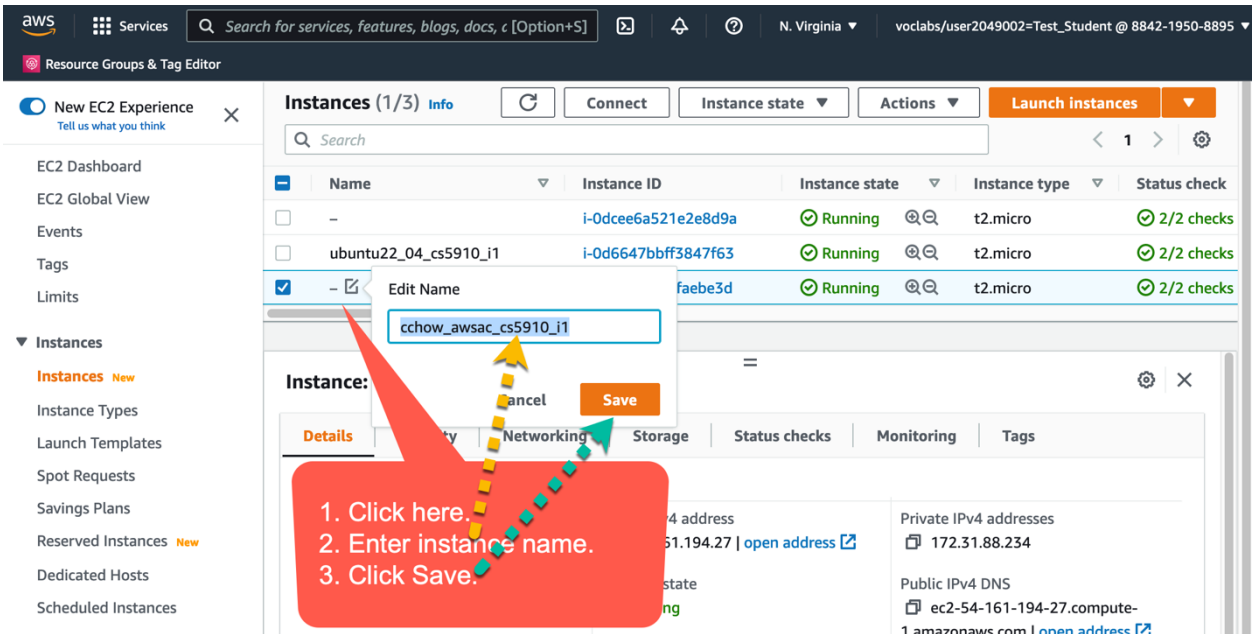

Figure 17. Edit name if needed.

2.2 Start and stop the instance.

When only the check box of one instance is checked, we can observe the public IPv4 address assigned to the instance in the current session. For example, in the dialog below, my jpan_awsac_cs5910_i1 instance is assigned with 54.85.96.25 public IP address.

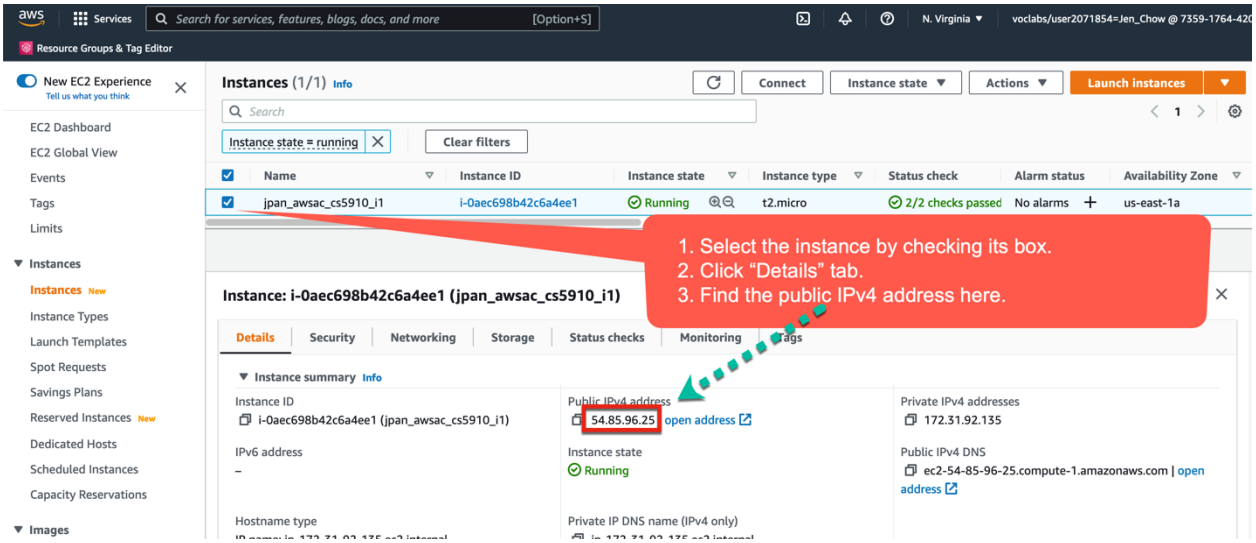

Figure 18. Find the instance and its assigned public IP address

Critical fact: I can use SSH terminal app to access my instance with this public IP. However this public IP address will be reassigned to other AWS instance, once we stop the instance. This is due to the lack of IPv4 addresses and the need to share precious addresses allocated to AWS.

We will use “Elastic IP address” AWS provides to associate an public IP address with our instance.

2.2.1. Stop an instance.

Select the instance. Then choose the “Stop instance” menuitem in the “Instance state” menu.

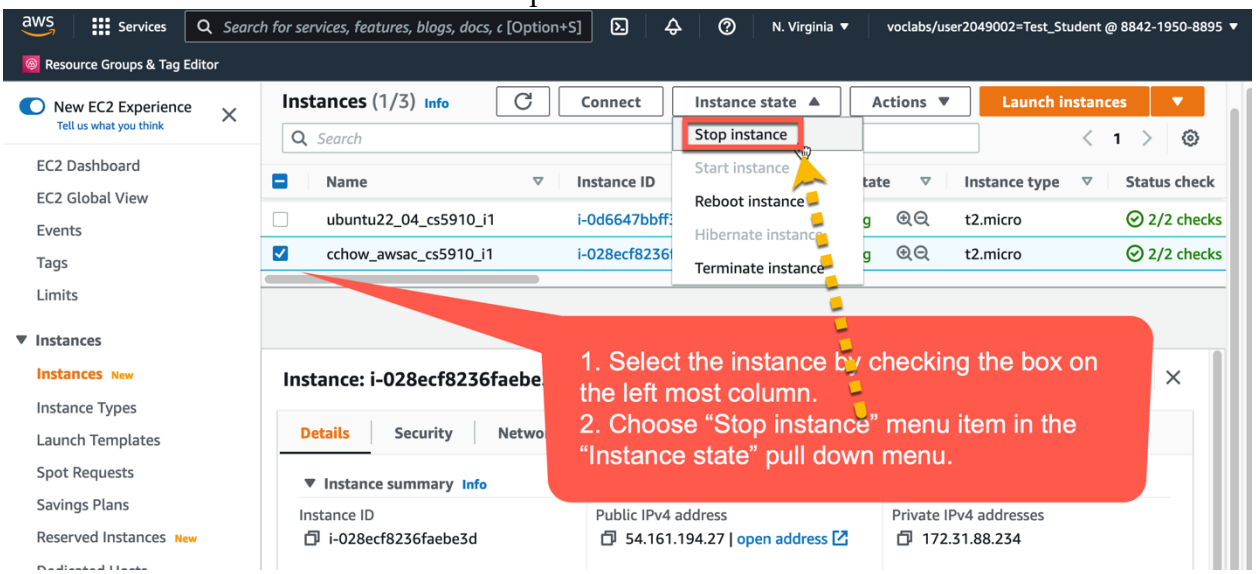

Figure 19. Stop an instance.

The instance’s state (4th column) will change from Running to Stopping, then to Stopped. Note that the public IPv4 address disappear (being put back to a pool for other instances).

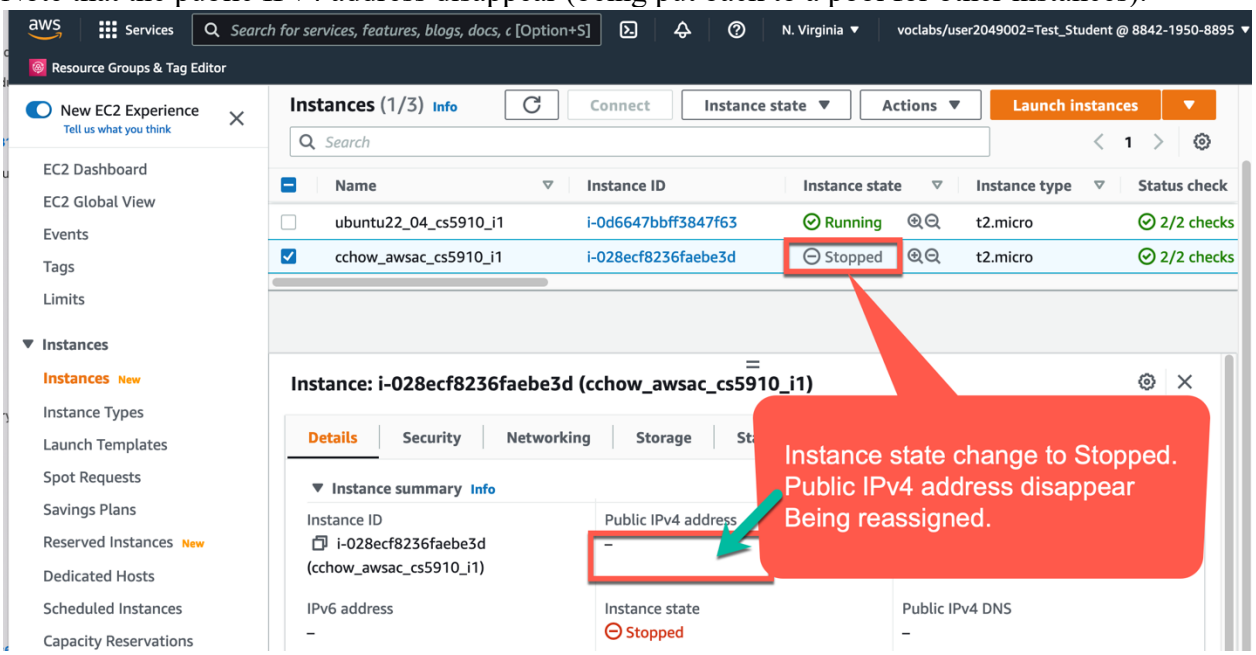

Figure 20. Dynamically assigned public IP address will be released.

Very unlikely the same public IP address will be assigned if we resume the instance operation. We will see in the next section.

2.2.2. Resume (start) a stopped instance

By selecting the instance and choosing the Start instance menuitem in the Instance state menu, we can start or resume the operation of an instance. The Instance state will change from Stopped to Pending, then to Running.

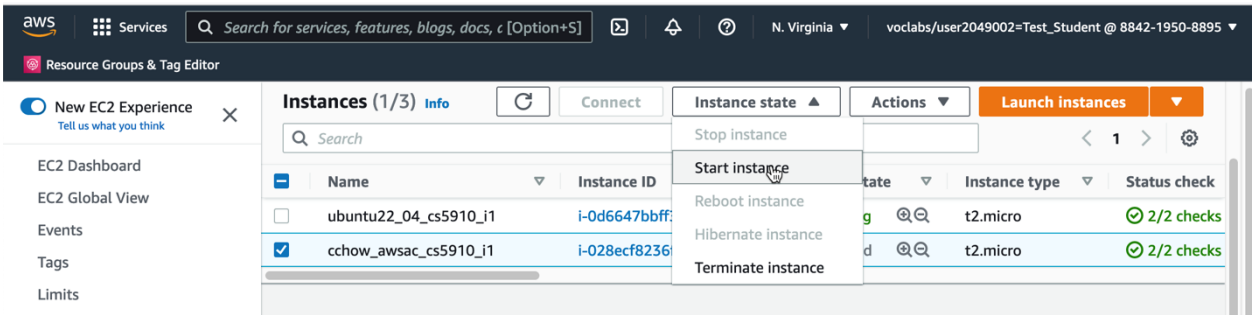

Figure 21. Resume your instance.

When it changed to Running state, we see a new public IPv4 address 34.226.141.236 is now assigned to our instance. It is different from the 54.161.194.27 that was assigned before.

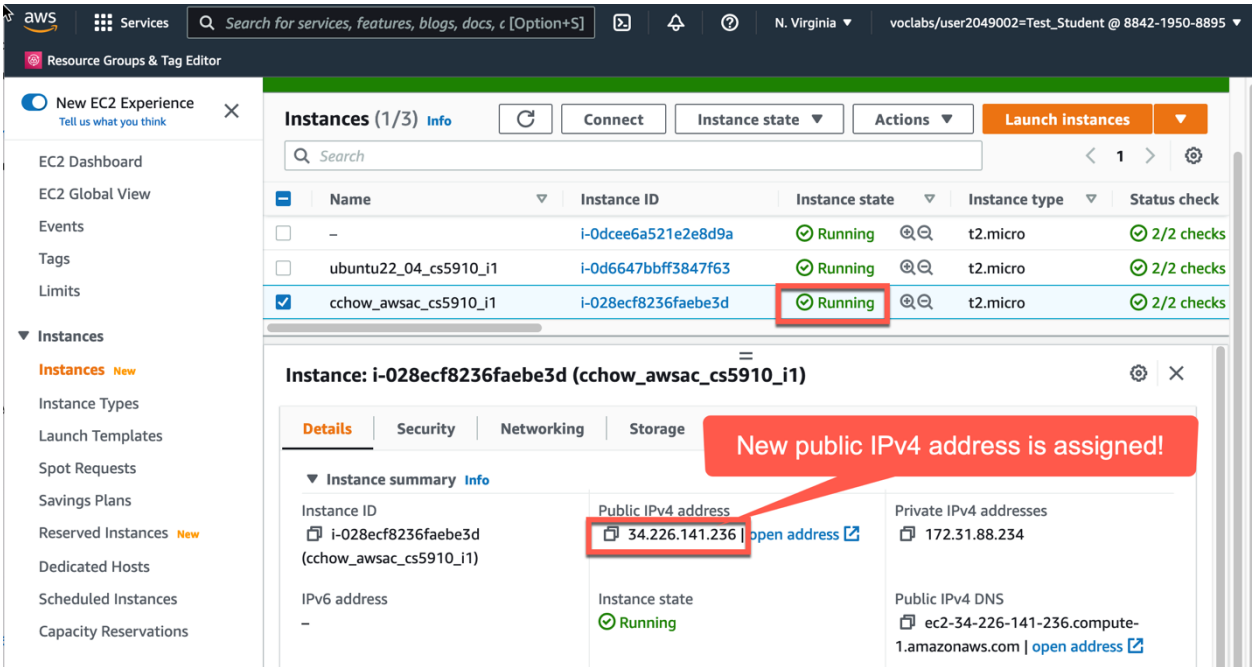

Figure 22. new assigned public IP address will be associated with the instance.

We now need to use this new IP address to connect to the instance.This is OK if the instance is only used as a client machine. It is troublesome if we need a "permanent" or static IP address to associate with our instance so that it can be served like a server.

With limited IPv4 addresses, AWS provides the Elastic IP address. It is like a static IP address associated with the instance. But if you are not running the instance, you will be charged for the

period you are not fully utilized this public IP address. Very unusual pricing scheme. Let us see next how to associate an Elastic IP address with our instance.

2.3. Associate Elastic IP Address with the Instance

To save IP address space, AWS will reclaim the public IP address of the instance once it is "stopped" or "suspended". Next time when we restart or resume the instance, AWS will assign a new public IP address for the instance. This often confuses the new users since they forget to check the new public IP public IP address in the detailed info panel for the instance. They thought the instance is not reachable for some other reasons.

To avoid such an inconvenience, AWS provides "Elastic IP Address" to be associated with the instance so that when the instance is resumed, it will have the same Elastic IP Address for accessing the instance. Note that when you are not using an instance with Elastic IP address (not intuitive), you will incur some charge, but normally it is much less than the suspension of the instance for a long period of time.

Follow the steps below to request an Elastic IP address and associate with the instance we just cloned:

2.3.1. Request An Elastic IP address.

Select Elastic IPs menu-item on the left panel of AWS EC2 dashboard. Click "Allocate Elastic IP address.

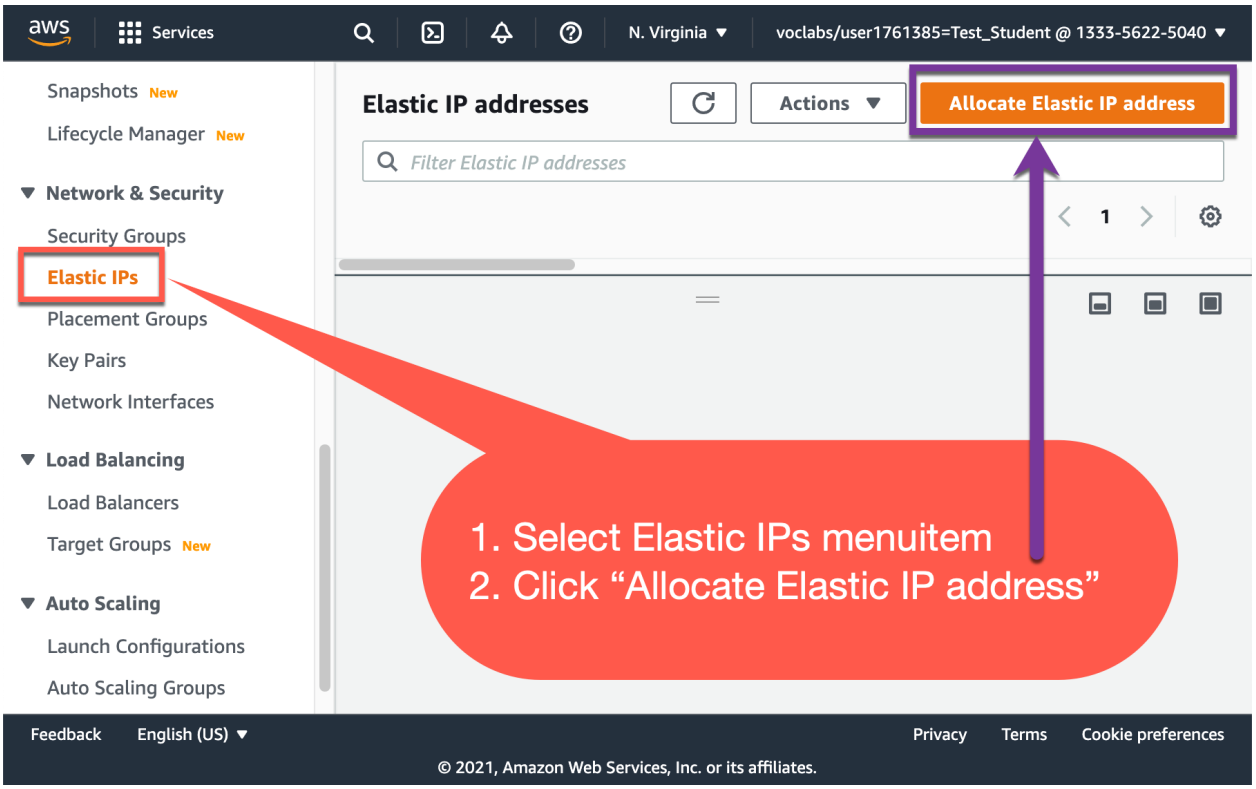

Figure 23. Request an Elastic IP address.

2.3.2. Confirm its allocation by clicking on the Allocate button on the lower right.

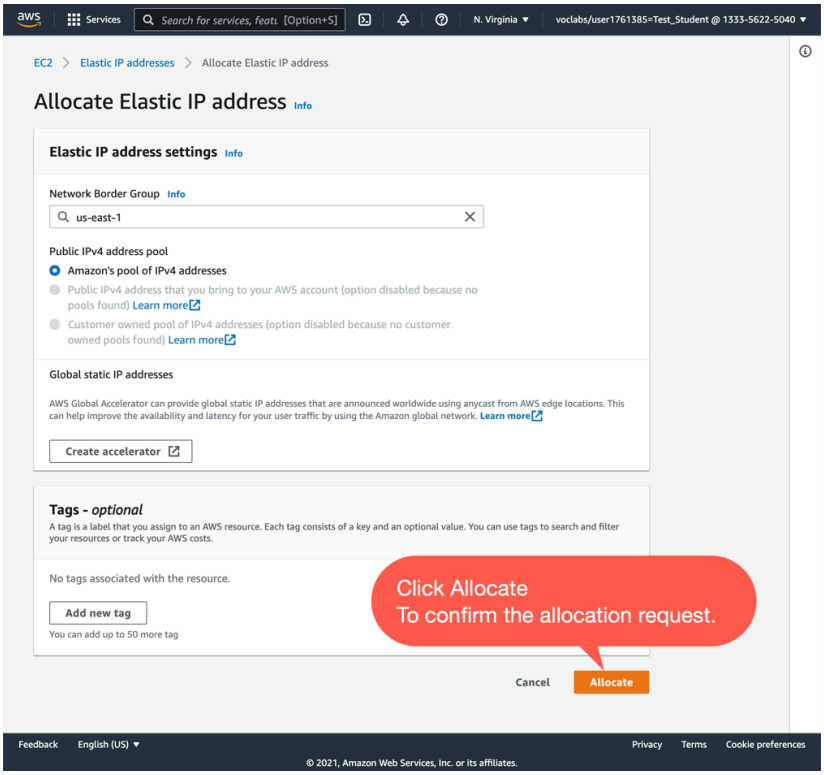

Figure 24. Confirm the allocation of Elastic IP address

2.3.3. Associate Elastic IP with instance.

Select the Elastic IP address just allocated. Click the “Associate this Elastic IP address” button or select that in the drop down “Actions” menu

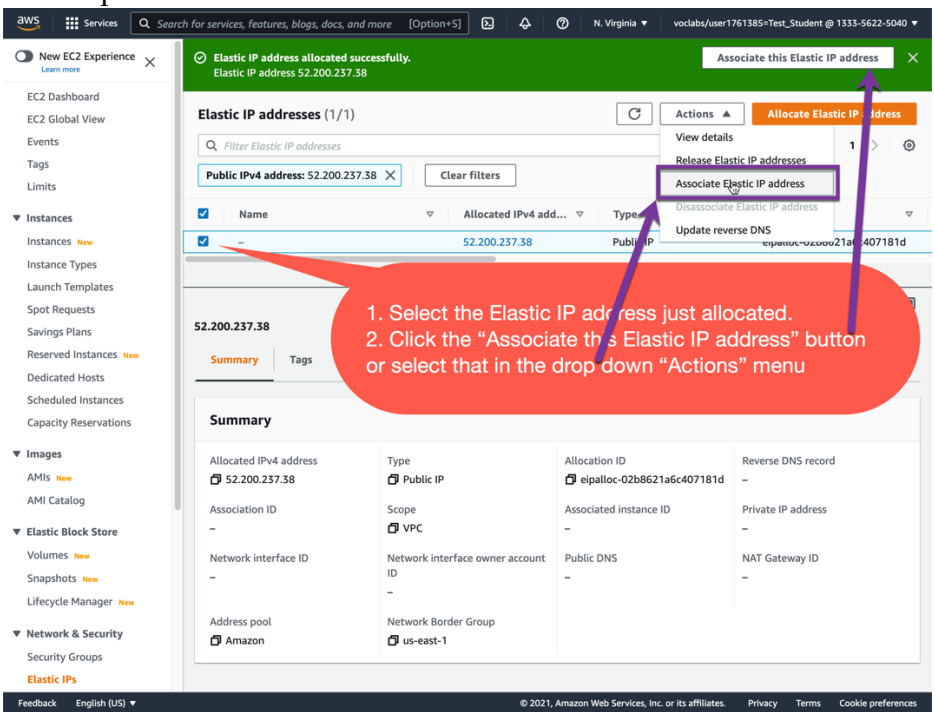

Figure 25. Associate IP address with an instance.

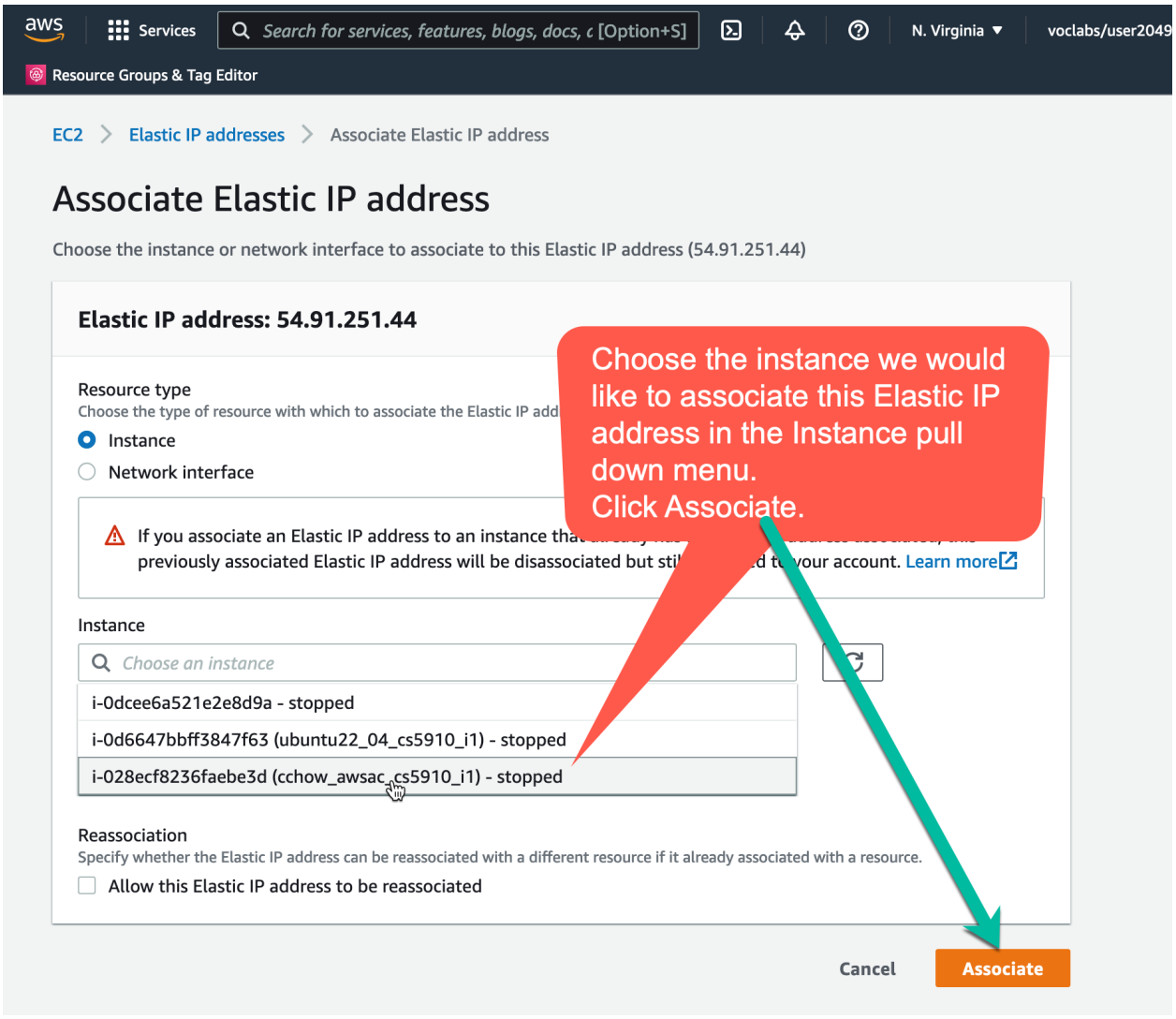

Figure 26. Select the instance to be associated with the current Elastic IP address.

2.2.4. Verify the instance is now assigned with the Elastic IP address.

Select the instance. The public IPv4 address now shows the associated Elastic IP address. You may have to refresh the info. Note that no matter whether the instance is running or stopped. The

details window shows the public IP address of the instance is the associated Elastic IP.

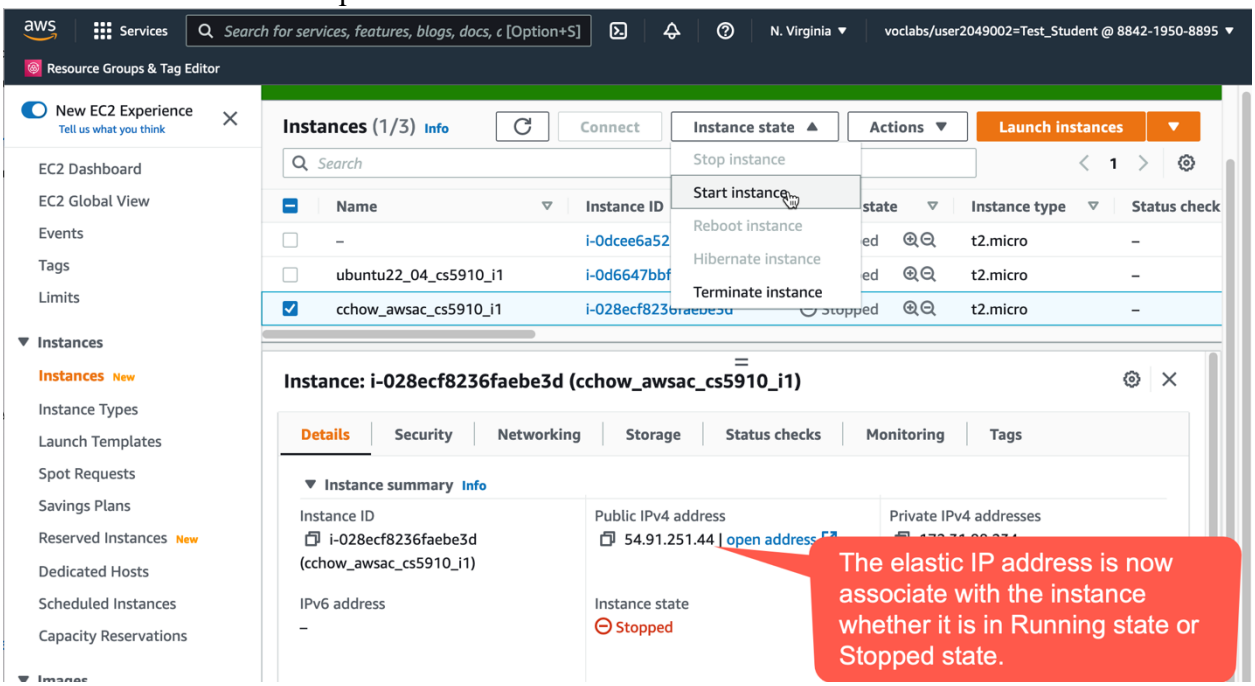

Figure 27. Verify the Elastic IP address just associated with the instance.

3. Access your AMI instance.

3.1 For mac or Linux users,

You can use ssh command in the directory containing your instance's private key.

```
ssh –i <privateKey>.pem ec2-user@<InstancePublicIPAddress>

Here <privateKey>.pem is the keypair name in the last prompt window of the instance
creation process;
<InstancePublicIPAddress> is the instance public IP address showing in the previous
diagram of the instance's info window;
–i indicates to the ssh client command we are using the specific private key file for
accessing the server.
```

Here the parameter right after -i option is the private key file name. Make sure the file access mode need to be change to 400 or r only by the owner, i.e., can only be accessed by you. You can use "chmod go-r <login>_awsac_cs5910a_key.pem" to remove group and other user access to your private key file. Without such change, the ssh will refuse to make connection. SSH client follows the new standard for security practice implementation.

Note that this SSH client command try to access the home directory of a user called "ec2-user" on a server or instance running SSH server daemon. In the AWS Amazon Linux 2 image, there is only one user ec2-user created as the first user. As a first user, ec2-user can use "sudo" precedes any privileged command to run typical system configuration operations like a root user.

Here is a session example with my related login info:

```
cchow@MacBook-Pro privateKey % ssh -i cchow_awsac_cs5910_pkey.pem ec2-
user@54.91.251.44
The authenticity of host '54.91.251.44 (54.91.251.44)' can't be established.
ED25519 key fingerprint is
SHA256:HmSpui0LrKnrH/3SV70o+hZAbVsJ8LwWH+qFM/L6v90.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.91.251.44' (ED25519) to the list of known
hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'cchow_awsac_cs5910_pkey.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "cchow_awsac_cs5910_pkey.pem": bad permissions
ec2-user@54.91.251.44: Permission denied (publickey,gssapi-keyex,gssapi-with-
mic).
cchow@MacBook-Pro privateKey % chmod 400 cchow_awsac_cs5910_pkey.pem
cchow@MacBook-Pro privateKey % ssh -i cchow_awsac_cs5910_pkey.pem ec2-
user@54.91.251.44

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
12 package(s) needed for security, out of 22 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-92-135 ~]$ sudo yum update
[ec2-user@ip-172-31-92-135 ~]$ sudo yum install nmap
[ec2-user@ip-172-31-92-135 ~]$ sudo nmap localhost

Starting Nmap 6.40 ( http://nmap.org ) at 2022-08-12 07:19 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000040s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
443/tcp  open  https
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
[ec2-user@ip-172-31-92-135 ~]$ mysql -u root -pcs00net
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 46
Server version: 10.2.38-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
```

```
MariaDB [(none)]> ^DBye
[ec2-user@ip-172-31-92-135 ~]$
...
```

In the above session, the ssh command detects the private key file was not protected against theft from local users and make sure that the user change its access rights to 400 before ssh can open with the private key. It enforces this by changing the file access rights and making sure only the owner can read the private key file content. Others or people in the owner's group cannot access or steal the private key. In a way this is the so called "Security by Design" paradigm. When design a security application, we should follow such a paradigm.

The other common system practice we observed above is that the system automatically check on the security patches that are available and remind the user to run "sudo yum update" to install the critical security patches. Yum is the package manager installed for the Amazon Linux 2 instance. Make sure you run "sudo yum update" as soon as you login.

We install nmap command and run it with localhost to show the ssh, http, https, and mysql (mariadb SQL Database server) servers are all running. You can also enter https://<yourInstanceIPAddress>/phpMyAdmin/ url where <yourInstanceIPAddress> needs to be replaced with the public IP address of your instance. You can then type in root as username and cs00net as password on the php web page with interface to the database server. It is a proof that your LAMP server package on your instance is working.

3.2 Access Amazon Linux 2 instance from Windows

The Windows users can use either PowerShell or Bitvise app to utilize the SSH access to the instance. You can download the Bitvise SSH Client app installer and install Bitvise app. It is available at https://www.bitvise.com/ssh-client-download. The current version # is 9.23. It provides a nice SFTP GUI for drag and drop files between remote server and your local client. It also does not require to go through the .pem to .ppk file conversion which is required by PuTTY app.

3.2.1 Using PowerShell to access your Amazon Linux 2 instance

Start PowerShell by typing "powershell" in Windows search box and click the Windows PowerShell app that shows up.

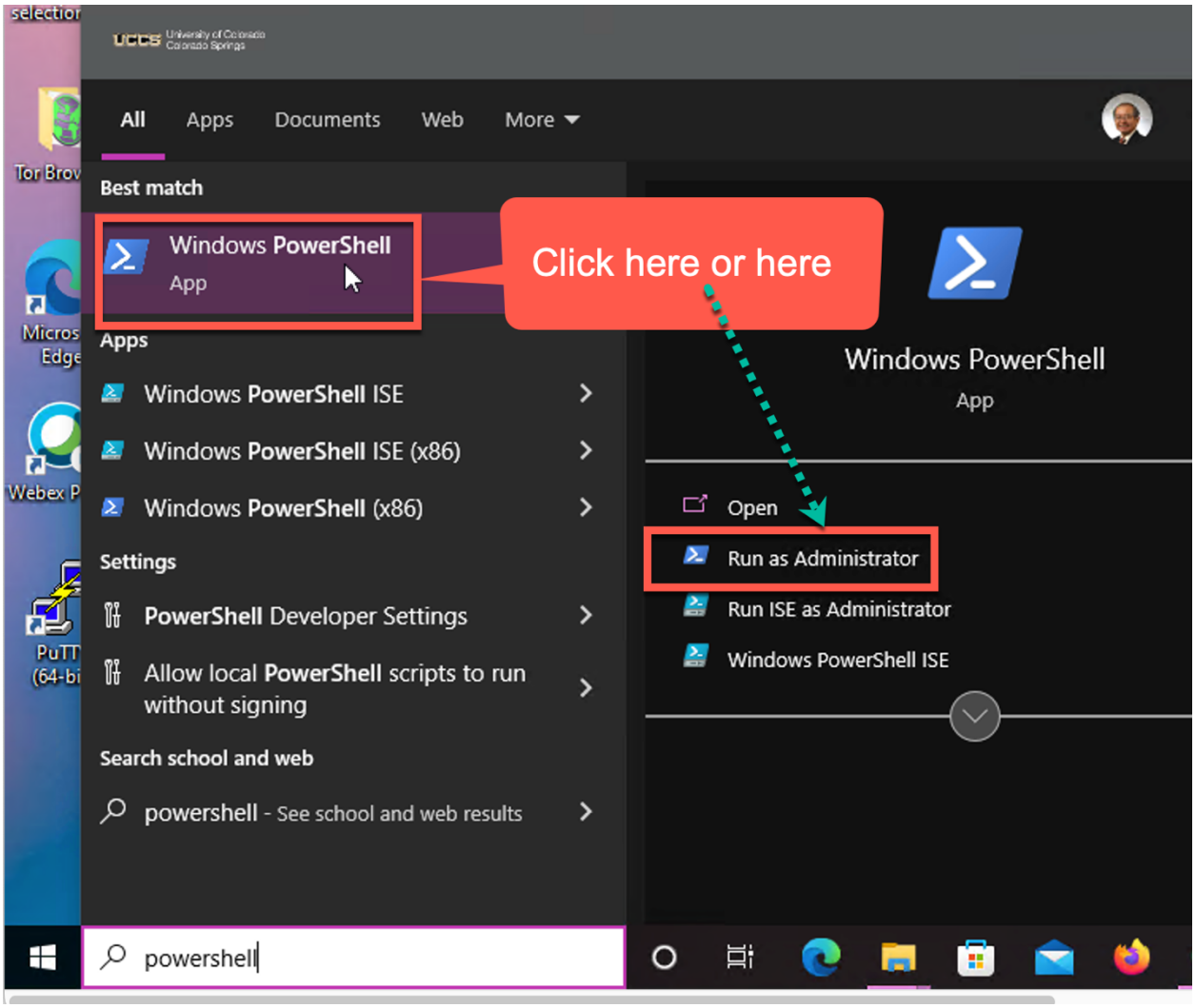

Figure 28. Start Powershell on Windows.

Find out where is the location of your downloaded private key. In my case, it is C:\Users\csnet\Downloads\cchow_awsac_cs5910_pkey.pem
We will use this filepath as the value for the -i option in ssh and scp command.

In PowerShell, we can use the ssh command for establishing a terminal session with our instance and use the scp command for copy files between the instance and the local client Windows machine. This is similar to what we use on Mac in Section 3.1.

Once the Windows PowerShell shows up, type the following command to establish a ssh session with your instance.

ssh -i C:\Users\csnet\Downloads\cchow_awsac_cs5910_pkey.pem ec2-user@54.91.251.44

The following image shows a ssh session is established by the PowerShell. Now whatever the command you type in after the [ec2-user-user@ip-172-31-88-234 ~]$ prompt becomes a Linux shell command. Here it shows the execution of ls command.

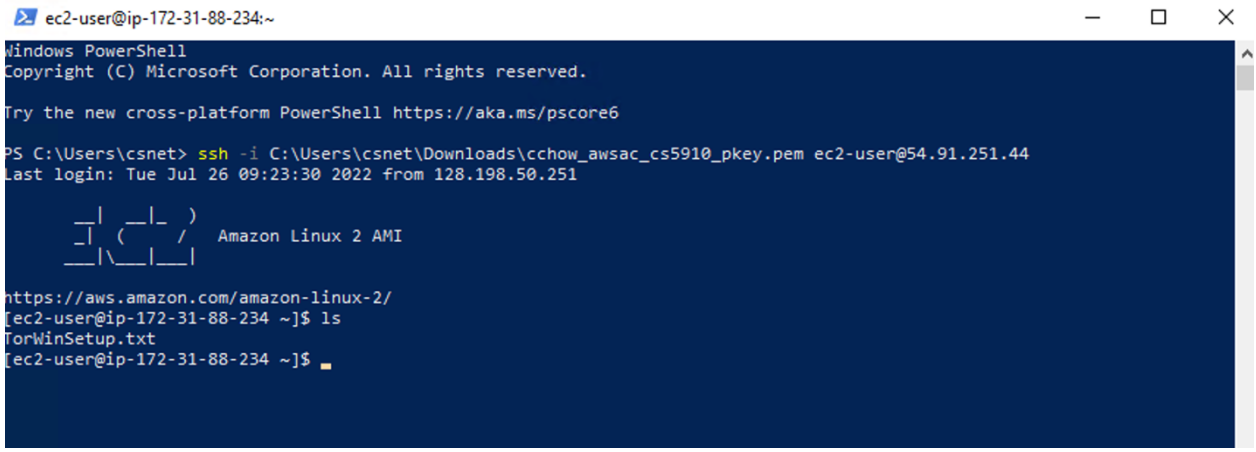

Figure 29. ssh command to create a session on our instance.

You hit control-D to logout from the ssh terminal session.

The scp command allows you to copy files between your instance and the local Windows client. It has the syntax of

scp -i <privatekey filepath> src dst

where src is the source of the document(s) to be copied from.
dst is the source of the documents(s) to be copied to.
<privatekey filepath> specifies the private key related to the instance.

Note that either src or dst can be a local Windows filepath or a remote instance filepath.
The remote instance filepath has the format of
<login>@<instanceIP/DNSname>:<remoteFilepath>
in our case the first part before : is ec2-user@54.91.251.44
the second part after : is whatever the local file path on the instance.

The following command copies a test.txt file in your local Windows client current directory to /home/ec2-user/ of an instance with IP address 54.91.251.44. Here . denotes the home directory of the ec2-user, i.e., /home/ec2-user/ is the destination directory to receive test.txt.

PS C:\Users\csnet> scp -i C:\Users\csnet\Downloads\cchow_awsac_cs5910_pkey.pem test.txt ec2-user@54.91.251.44:.

The following command copies a remote file test2.html in the home directory of ec2-user on the instance with 54.91.251.44 as public IP address to the current Windows directory.

PS C:\Users\csnet> scp -i C:\Users\csnet\Downloads\cchow_awsac_cs5910_pkey.pem ec2-user@54.91.251.44:test2.html .

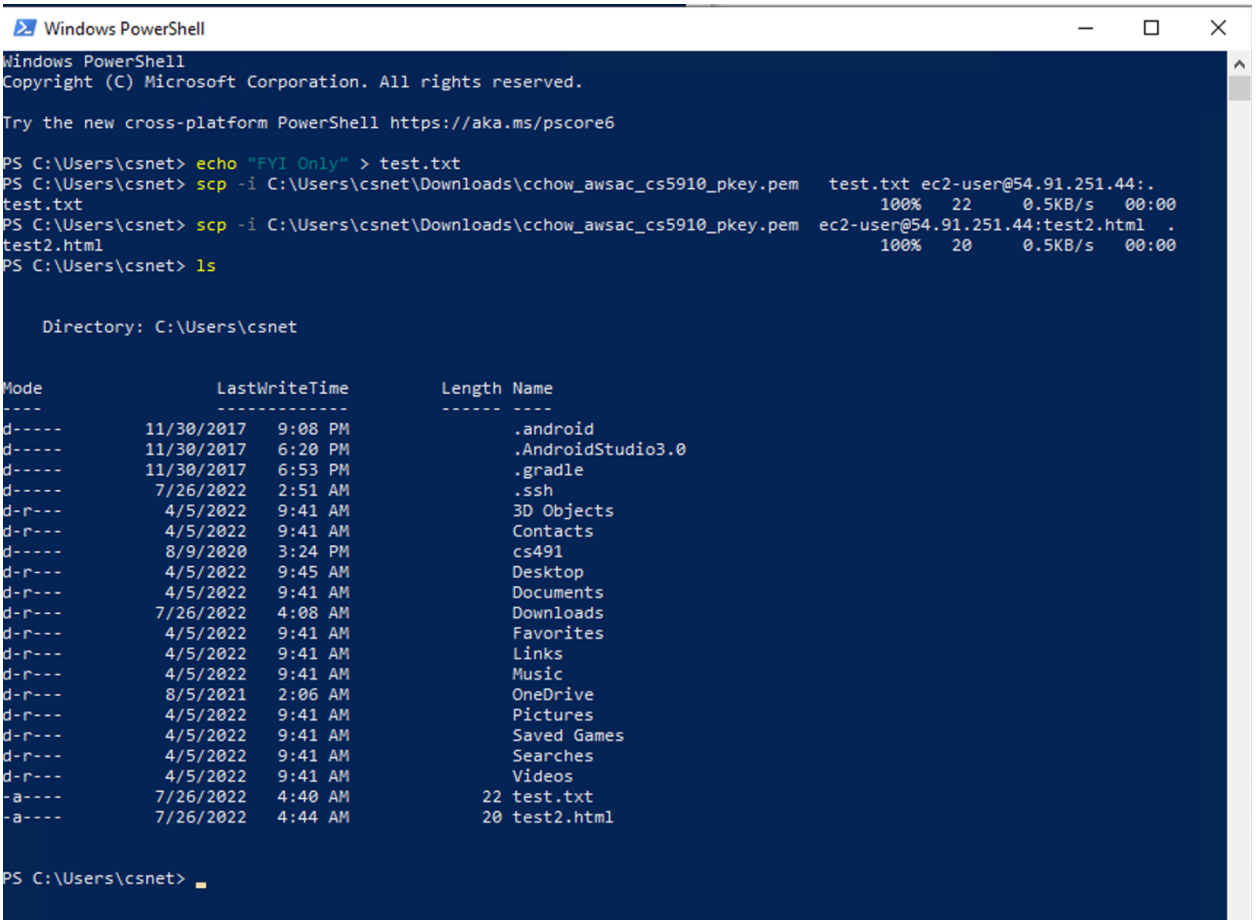

Figure 30. use scp command on Powershell to copy data among machines.

Note that you can only run these two scp commands on your local Windows client. You can only run similar scp command on your instance when you setup the sshd server on your local Windows client to receive files.

Remember to run the four sudo commands and mysql command in page 24 on your instance as soon as you login to your instance.

3.2.2 Install Bitvise SSH Client

After download and install the Bitvise app, you should see the following app installed on the upper left corn or the desktop.

Step 1. First import the private key into the bitvise app using its Client key manager.

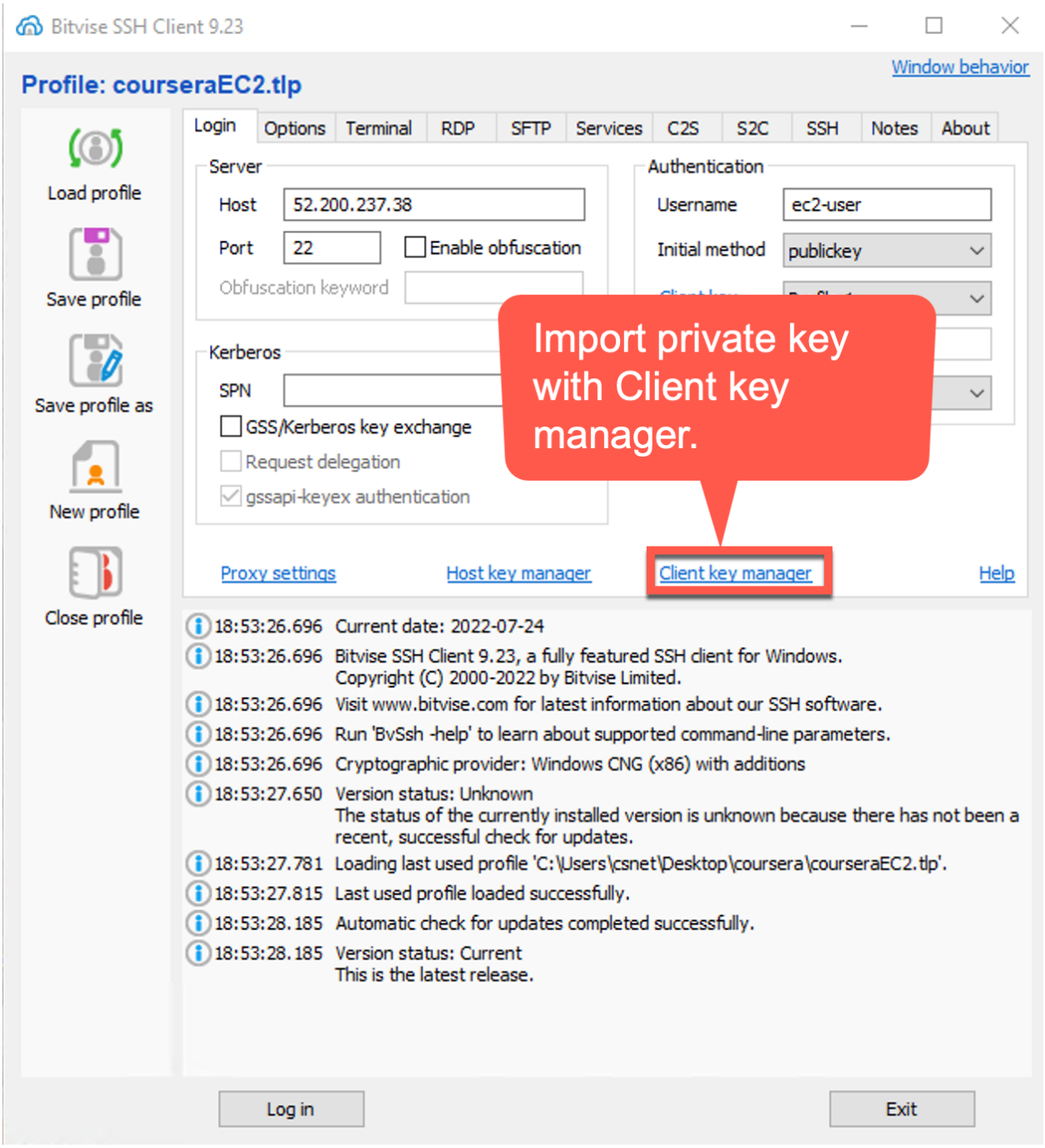

Figure 31. First import the private key into the Bitvise app using its Client key manager.

Step 2. Click Import button.

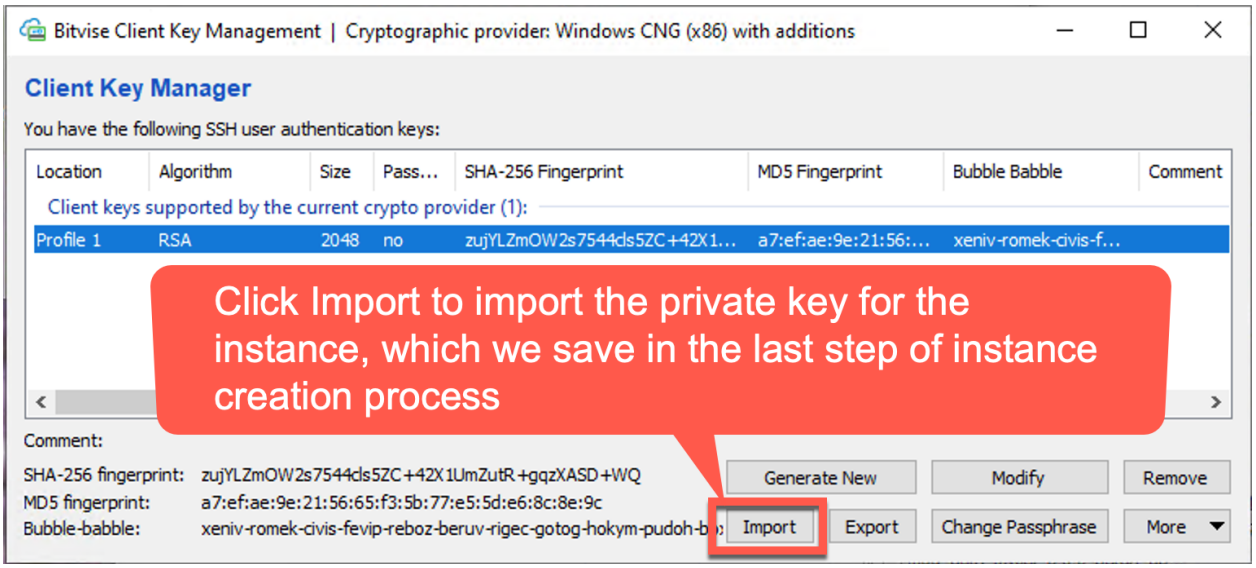

Figure 32. Click Import button to import the private key.

Step 3. Look for private key file.

Change the file type to all file (*.*) in order to reveal our private key in .pem file format. Note that the Bitvise Keypair Files (*.bkp) type will not show the .pem file type which is used to generate the private key during the last step of instance creation process.

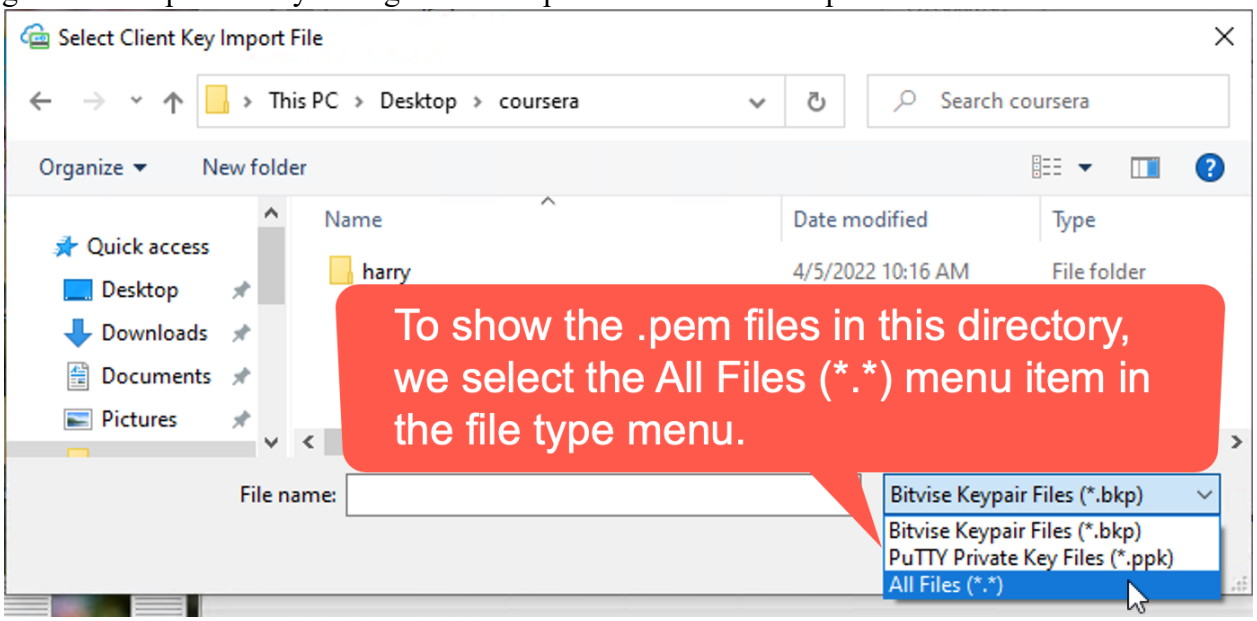

Figure 33. Change file type to All Files (*.*) to reveal our private key in .pem file format.

Step 4. Select our private key and Click Open.

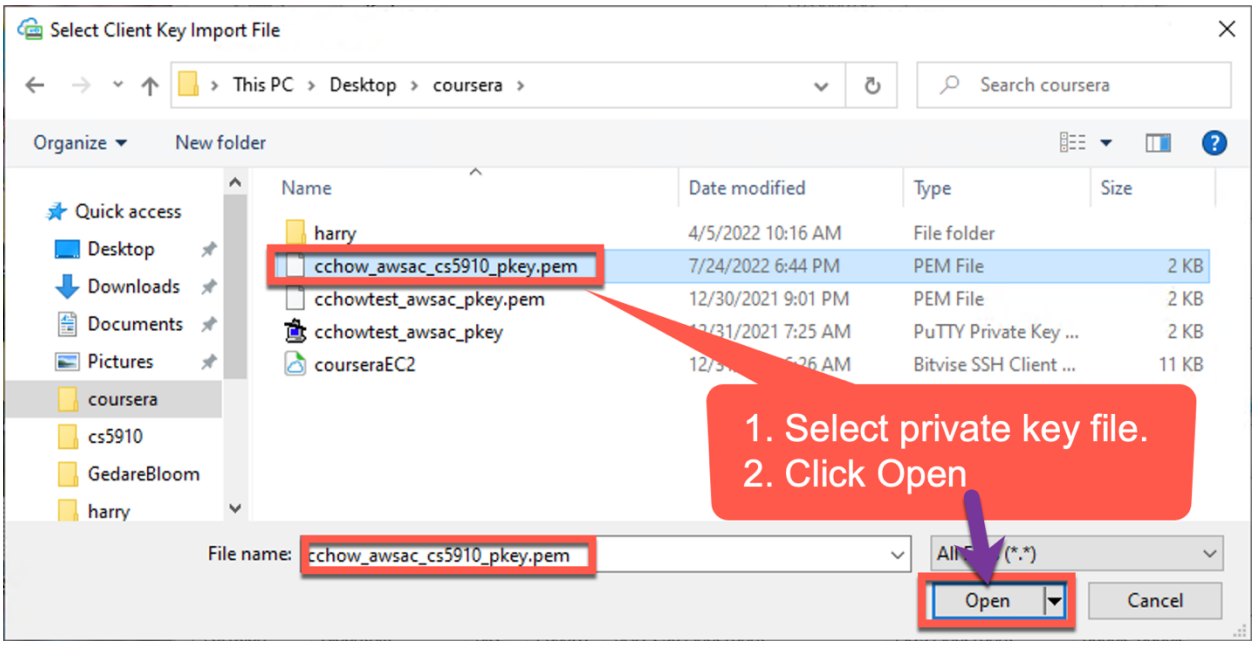

Figure 34. Select private key in .pem file and click Open.

Note that in Figure 34, we use cchow_awsac_cs5910_pkey.pem as an example.

Step 5. Import the private key

Once we click "open" on the selected .pem file, it will be converted to .ppk file format and show in the following window with potential local. Normally it is saved in Location Profile 1. Here it is saved in Location profile 2, due to previous saved private key.

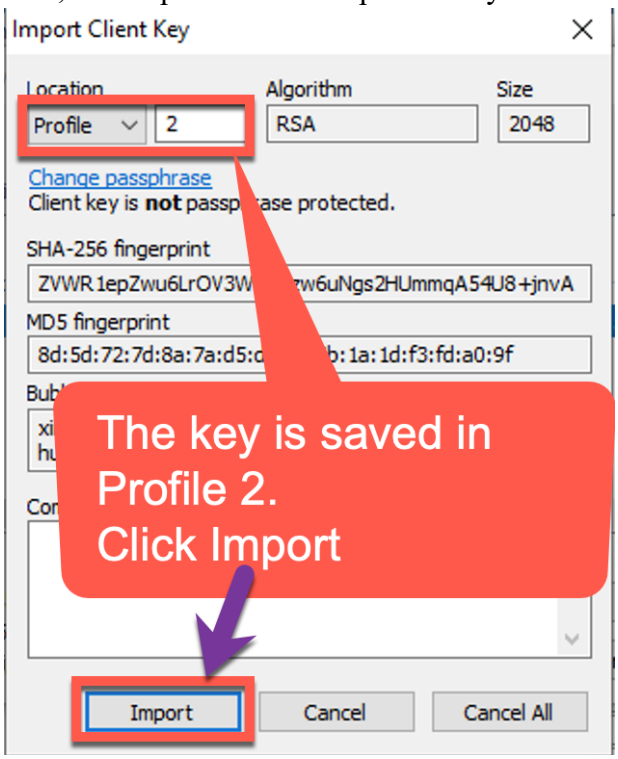

Figure 35. Save it in Profile Location key #2 by default and Click "Import".

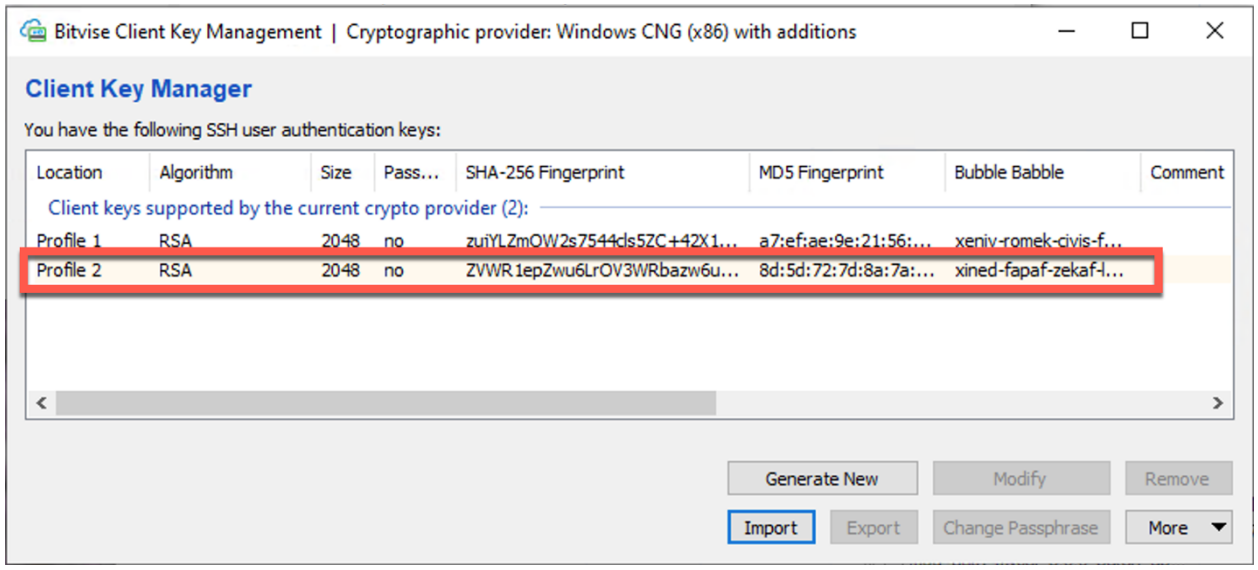

Figure 36. Client Key Manager display the imported Profile 2 key, our private key.

Step 6. Access the instance with Bitvise.
We need to specify the instance IP address, username, access method, the private key used for accessing the instance.

1. Specify public IP address of the instance

Make sure you use the current elastic IP address of the instance. In my case it is 54.91.251.44

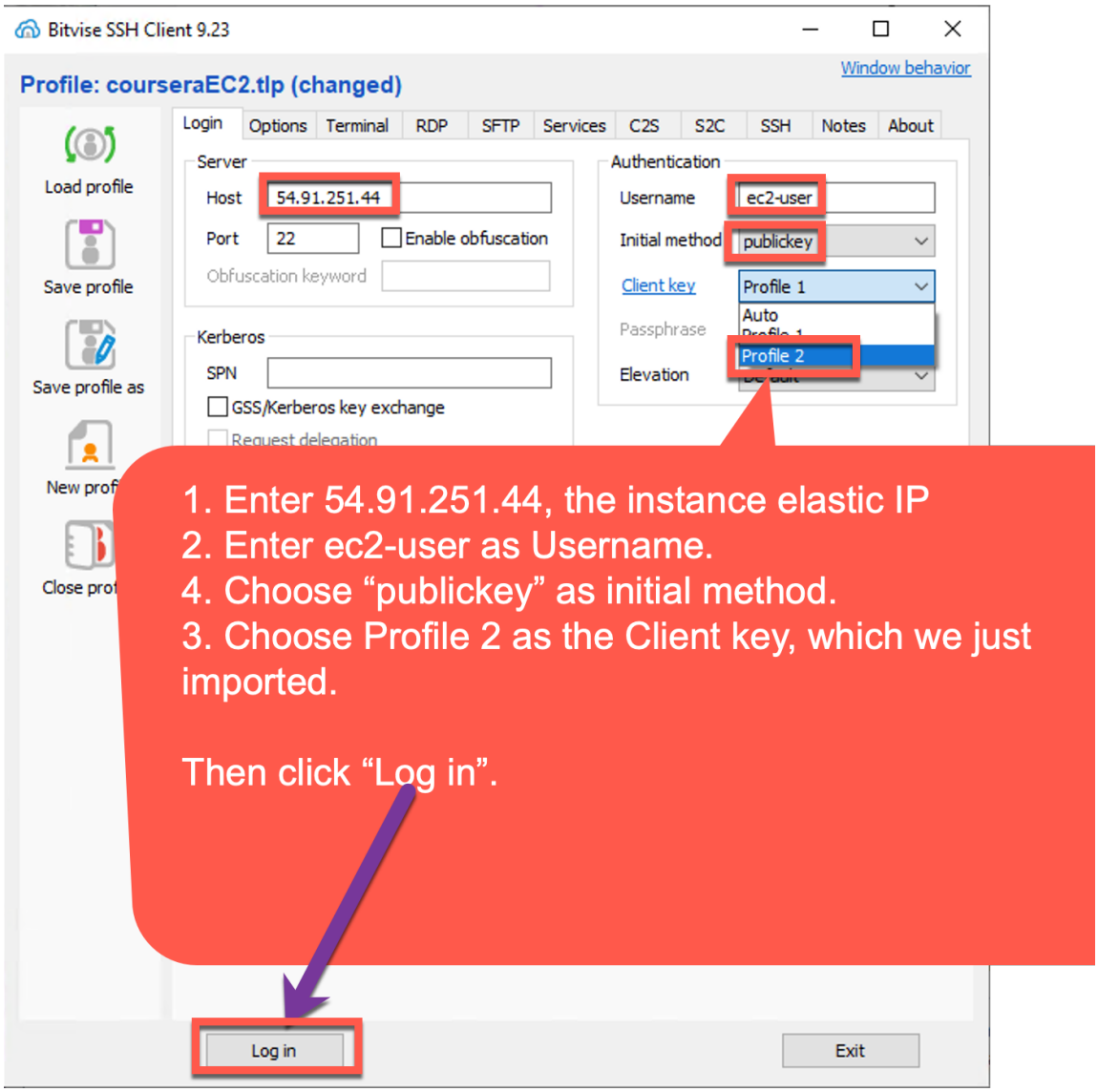

Figure 37. Bitvise profile setup.

2. Specify ec2-user as Username

Use “ec2-user”, not the “root” for accessing the AWS EC2 instance, since its sshd is configured to be refusing root direct login and there is initially only this single “first” user created.

3. Choose publickey as Initial method

Here we are telling the bitvise SSH client that in this SSH authentication protocol, the SSH Server will use the public key saved in the /home/ec2-user/.ssh/authorized_keys file for verifying the security token generated by the SSH client where a known string

is encrypted by the private key and the public key crypto algorithm.

4. Choose Profile key #2 saved in the Bitvise SSH Client as private key for this authentication.

5. Click Log in

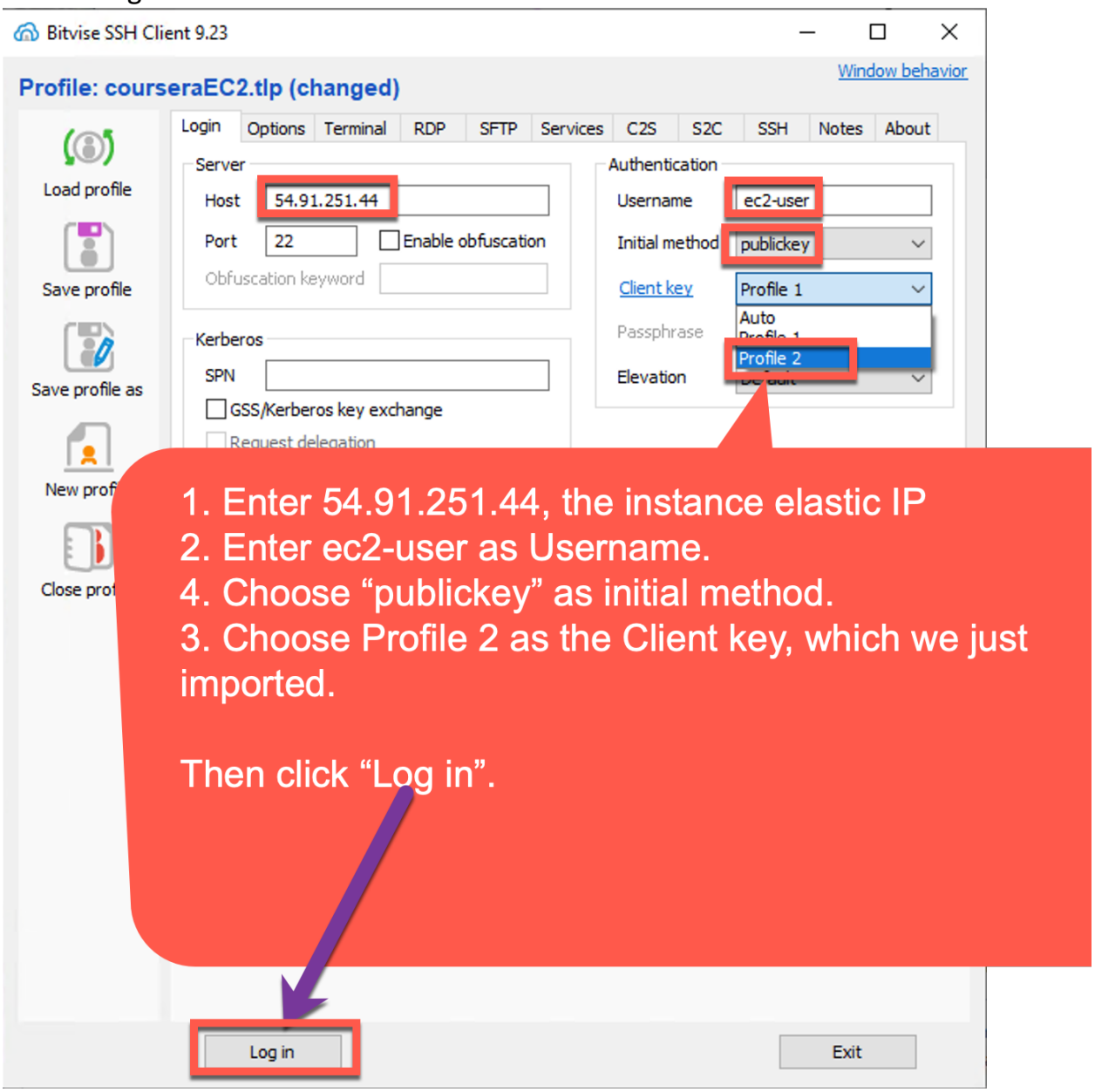

Figure 38. Specify parameters for accessing the SSH daemon on the instance server.

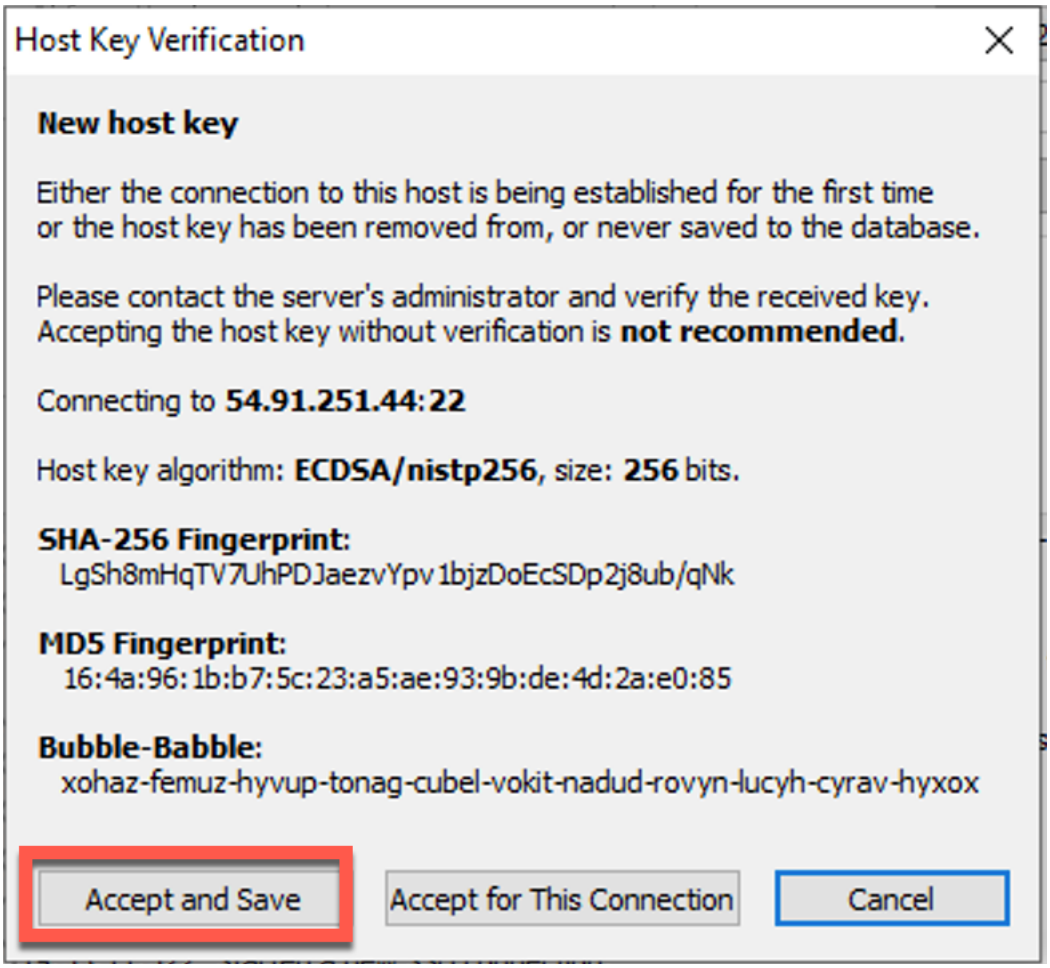

Figure 39. Accept and Save the new host key.

Step 6a. Deal with Connection Error.

If you observe the following error msg in Figure 7c, there could be several reasons:

1. Server instance does not start or got stopped by AWS Academy when the session exceeds certain time limit. We need go back to "start instance" using the AWS Console.
2. Your client machine was assigned with newer public IP address since last connection to the Internet through your ISP. We need to fix the problem by the source list of the instance's security group using MyIP. See Figure 7d for the steps to reset the My IP as the source list for the security group.

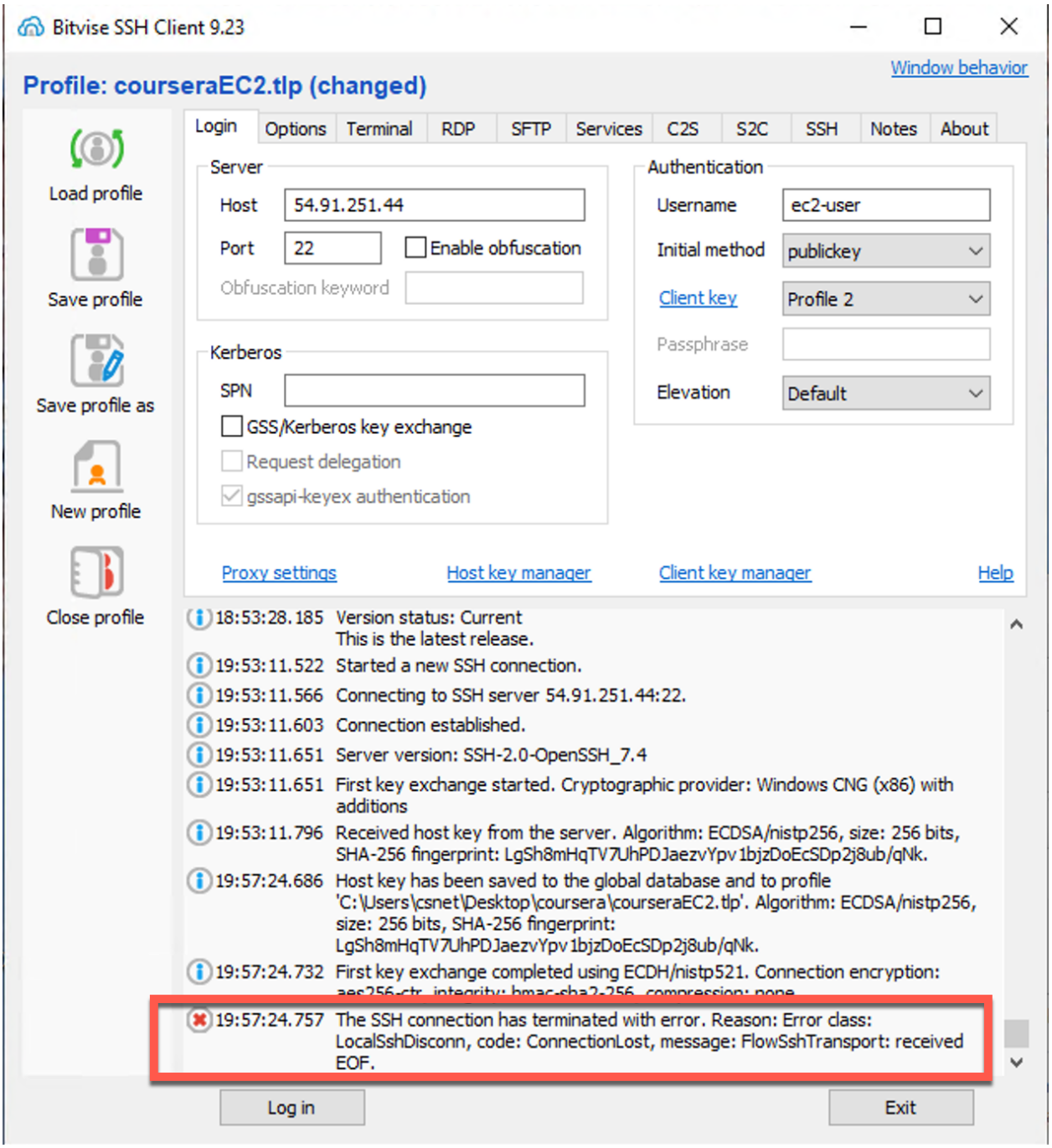

Figure 40. Failed connection due to new client IP assigned by your ISP.

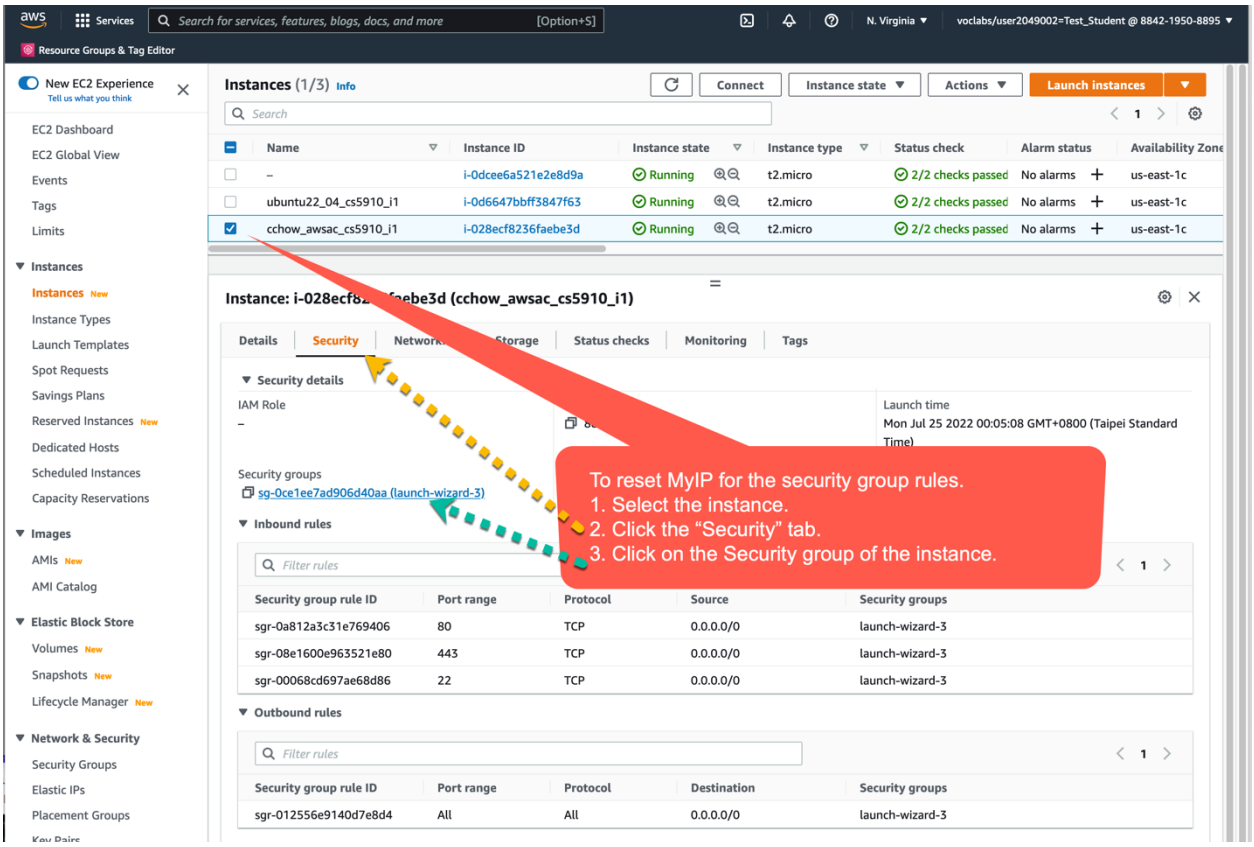

Figure 41. Steps to access the security group information.

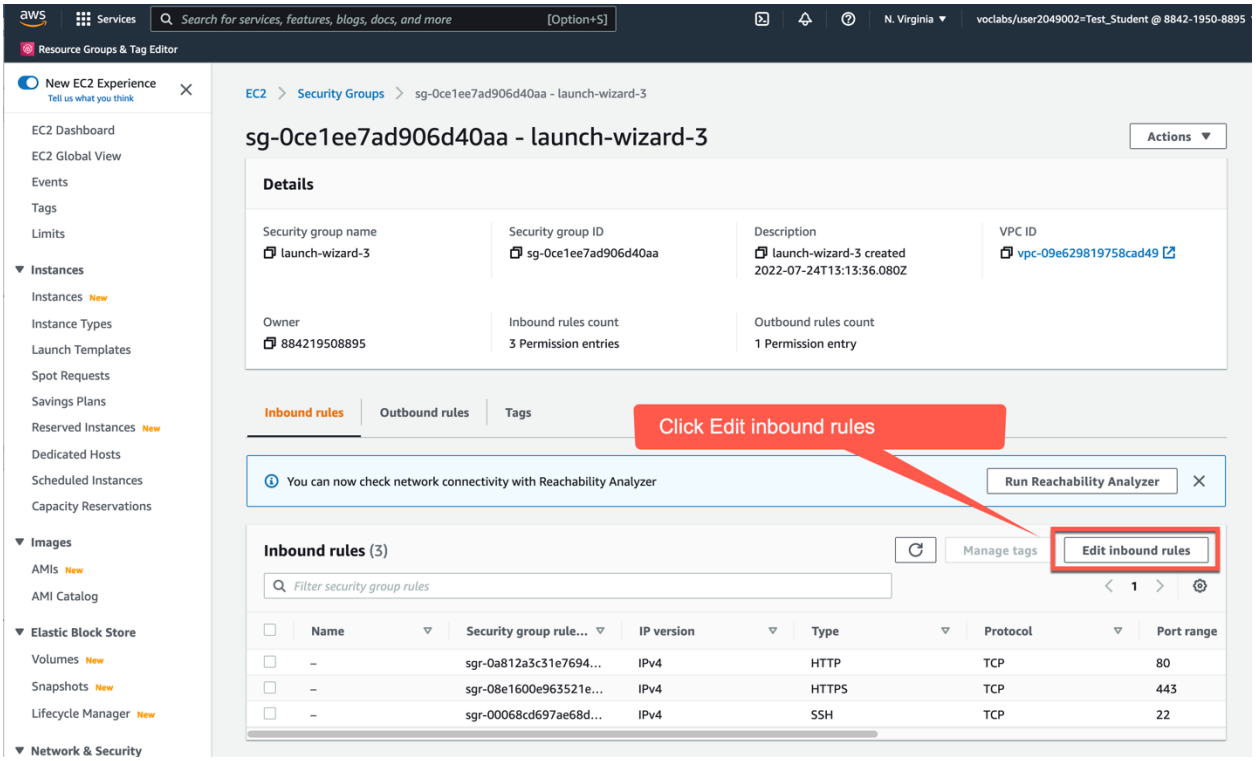

Figure 42. Edit inbound rules.

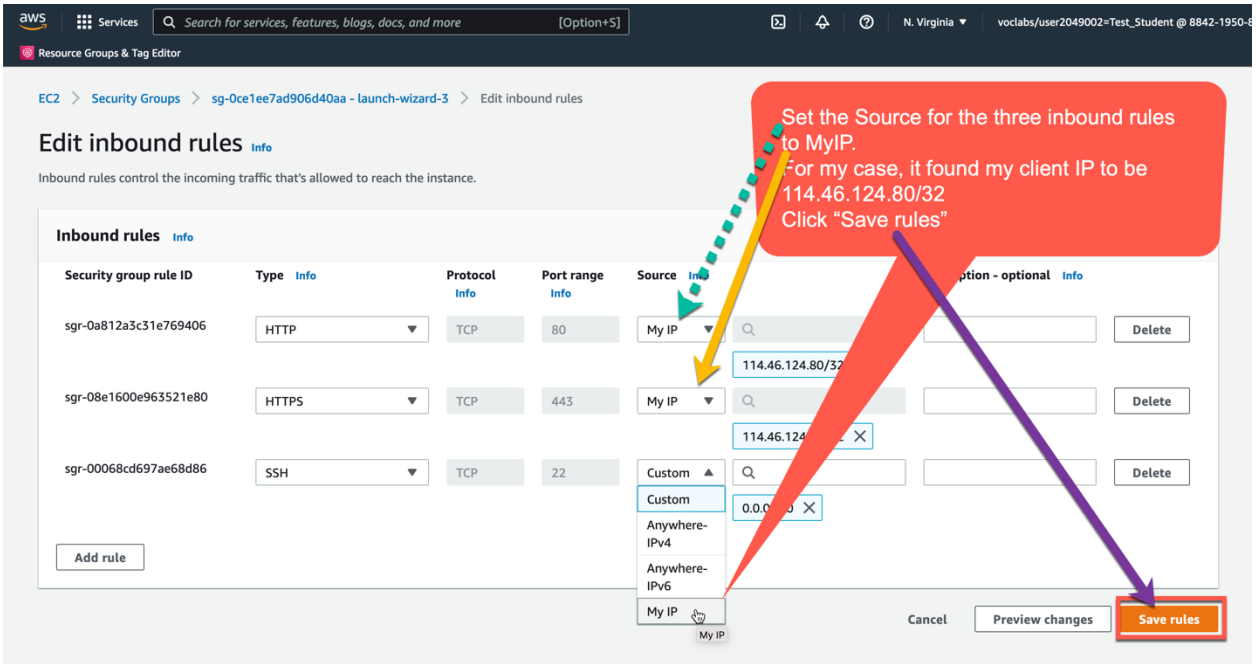

Figure 43. Reset MyIP for the source list of the security group.

Step 6b. The bitvise SSH xterm client and SFTP client are launched,

Finally the Bitvise app launches two tools: Bitvise SFTP Client and Bitvise xterm Client. We use Bitvise SFTP Client for dragging and dropping files between SSH client and SSH server and use Bitvise xterm Client for entering/executing commands on a remote terminal session on the instance.

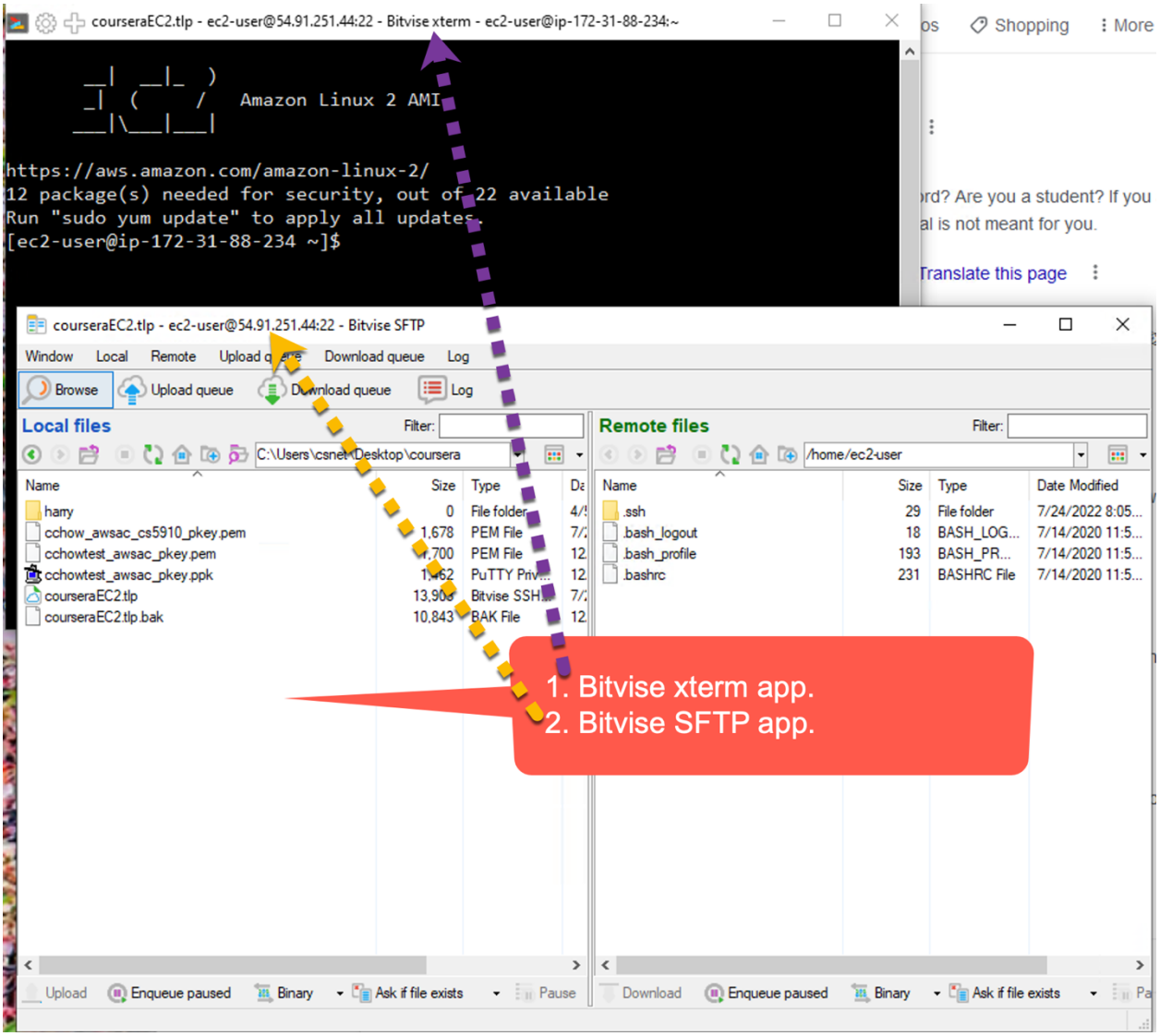

Figure 44. Bitvise SFTP Client and Bitvise xterm Client.

Remember to run the four sudo commands and mysql command in Page 24 on your instance as soon as you login to your instance.

Step 8. Save the SSH configuration as a profile for future reference.

Click "Save profile" and enter CourseraEC2 as profile name. Next time we start bitvise, it will remember the current profile. In we have switched to different profile, we can also click Open profile to bring back the SSH settings for accessing the instance. Click Login to verify if it works.

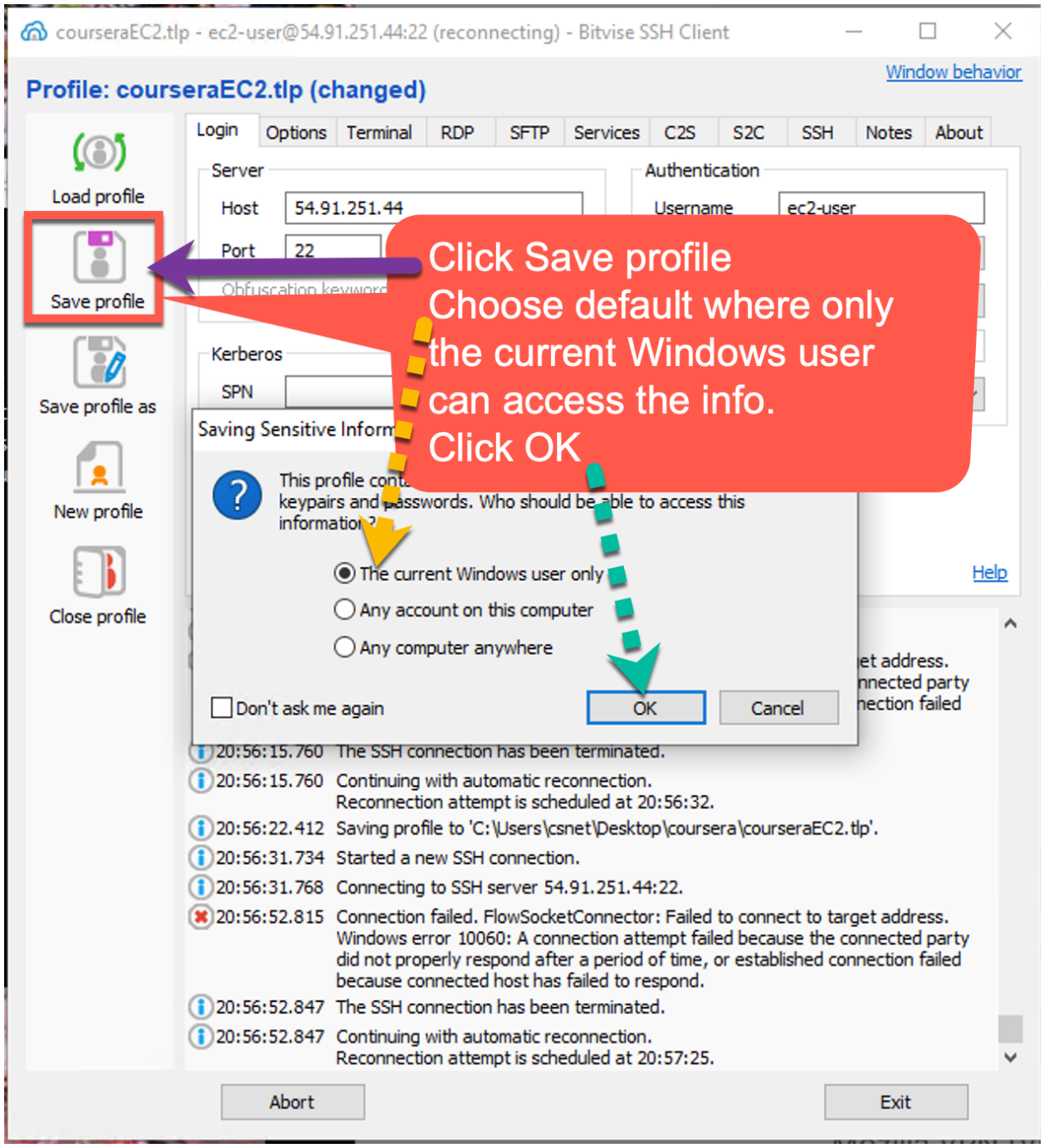

Figure 45. Save profile for future reference.

You can also “Save Profile as” as a different file name and to a different directory.

3.3. What you do when you get “failed to connect” message?

When your SSL terminal client failed to make connection to your AMI Linux server instance, there could be several reasons and some can be easily solved, others may take systematic diagnoses:

Possible Reason 1: You may not set the Source List of the security group of your instance properly.

The common situation is that we moved our client machine to a different subnet or location and was assigned with a different public IP address. In that case we need to reselect the “My IP” in the source column of all the related security group rules, so that the instance will accept the new connections coming from that new client location.

The security group specifies the firewall rules that govern the incoming or outgoing connection. The source list of the incoming firewall rules is particularly important and need to be configured correctly. It should include the current assigned IP address of your client machine. Without that, any connection from the client will be rejected. In Page 11 of Section 2.1.6. Step 6. Configure Security Group, we show how to add the current IP address of your client to the allowed connected Source List. Perhaps you missed the step. Here is how to add it after the instance is already created and running:

Step a. Select the instance and choose its security groups.

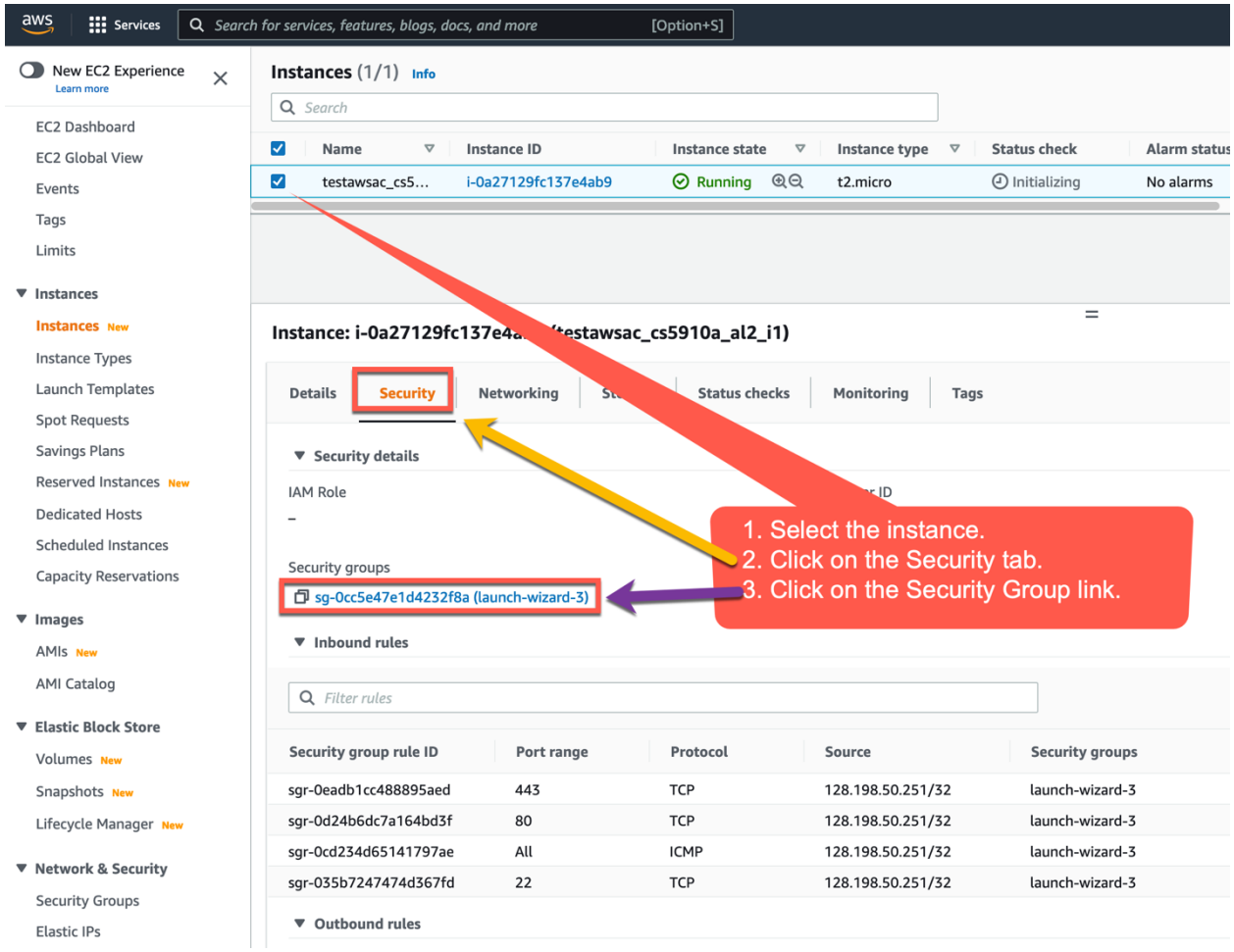

Figure 46. Access the security group specification.

Step b. Edit inbound rules.

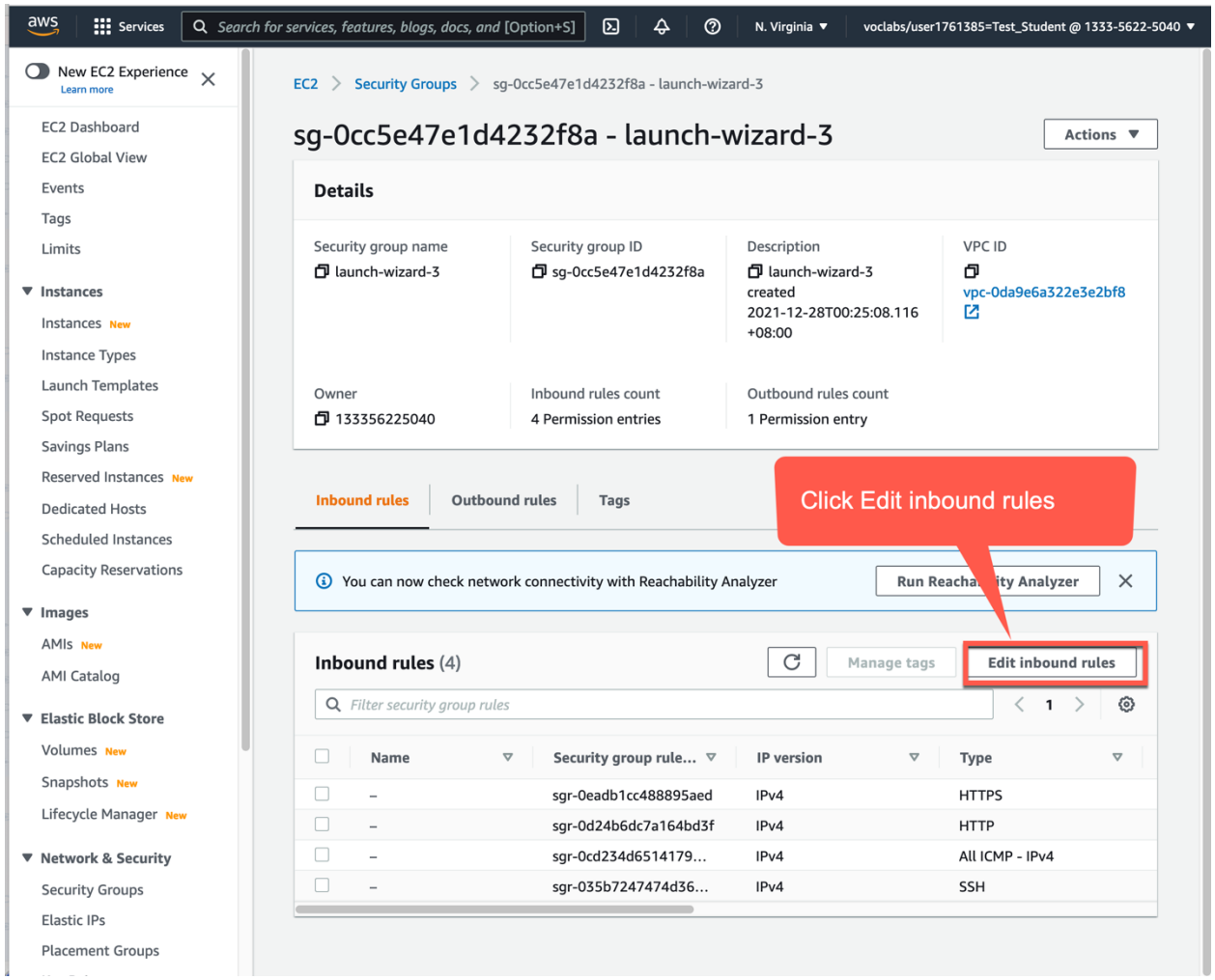

Figure 47. Edit inbound firewall rules.

Step c. Set new My IP

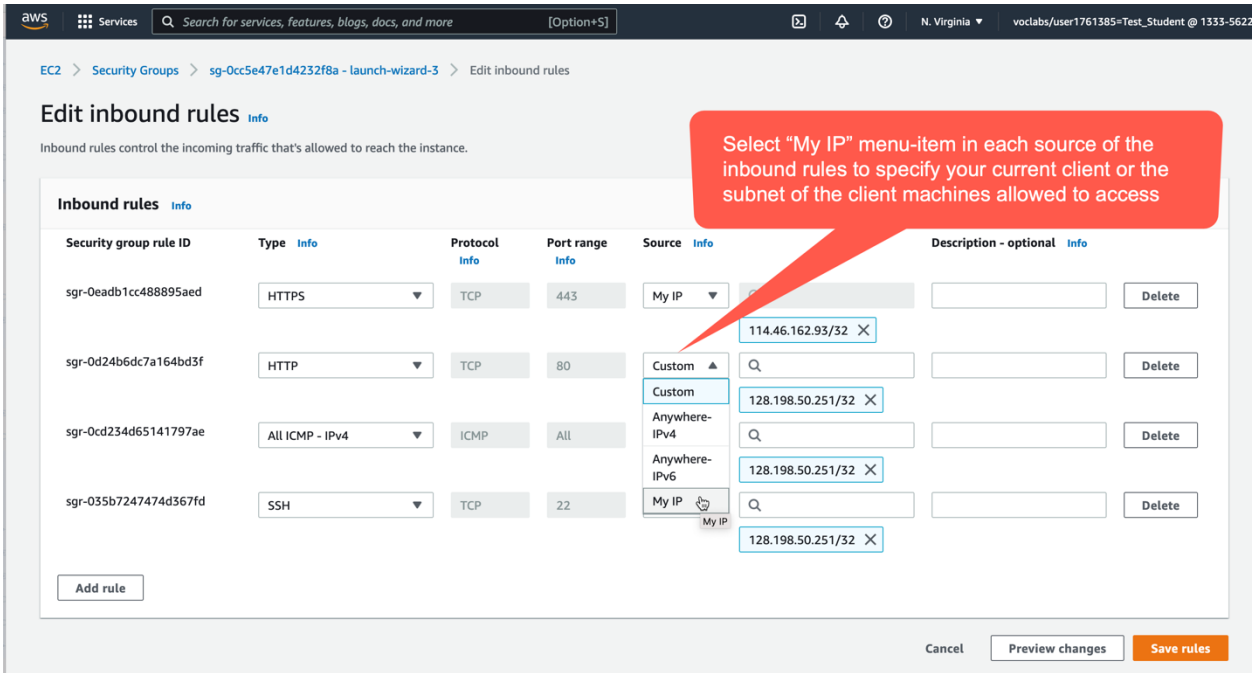

Figure 48. Set My IP on the source list of all rules to block potential hacking attacks.

Possible Reason 2. Use wrong private key or the app can not find the right private key. Check if the private key you configured is associated with the creation of the related instance. Often we found a wrong private key is used, when there are multiple private keys for multiple instances. For Mac or Linux client, when you use ssh command with -i option, make sure you are in the right directory that contains the related private key.

Possible Reason 3. There is potential Internet outage between you and the AWS region. Make sure you get response from a server in the same AWS region of your instance. In our case, for learners using AWS Academy free service, they can use command "ping -c 2 rds.us-east-1.amazonaws.com" or access web page https://rds.us-east-1.amazonaws.com and see if you get response. If not, you can systematically check if it is the network problem between your machine and the AWS region. Starting from ping your local residential gateway such as 192.168.0.1, and see if you can reach a known internet site you often visit.

4. Optional Exercise: Install LAMP Server Package

Many of our cybersecurity exploit examples can be illustrated with a server system installed with Linux Apache MySQL PHP (LAMP) server package. You can clone an instance from an update-to-date Amazon Linux 2 instance. Follow the same steps in Section 2.1, but in page 7 Section 2 Application and OS selection. Select the default Amazon Linux 2 image with Kernel 5.10 instead of choosing the coursera_cs5910a_im1 image. Finish the cloning process and associate an elastic IP address with your brand new instance.

Next follow the steps in the web page titled "Tutorial: Install a LAMP web server on Amazon Linux 2"

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-lamp-amazon-linux-2.html
to set up LAMP servers on our new instance. We will remote login to our newly cloned AWS instance to run the related system configuration commands. It will take about 25 minutes to complete the whole process, including the installation of phpMyAdmin to manage the MySQL server through php web pages.

Note that in the above web page, it does not provide detailed info in setting up and running a httpd server that supports for HTTPS (HTTP Secure) which protect your data with SSL/TLS encryption. To add that support, please follow the instruction in the web page
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/SSL-on-amazon-linux-2.html

Use the following yum command to install the mod_ssl package for HTTPS.

```
[ec2-user@ip-172-31-84-242 ~]$ sudo yum install mod_ssl
...
```

Here we run the four commands in the instance to create and set up certificate and private key for the web server.

```
[ec2-user@ip-172-31-84-242 ~]$ cd /etc/pki/tls/certs
[ec2-user@ip-172-31-84-242 certs]$ sudo ./make-dummy-cert localhost.crt
[ec2-user@ip-172-31-84-242 certs]$ sudo vi /etc/httpd/conf.d/ssl.conf
```

Using the editor we comment out Line 107 by adding # as the first character in that line, similar to the one below.

```
#SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

The reason we do that is the localhost.crt combines the certificate and private key created using the make-dummy-cert. In this simple set up, they are combined in a single file. Normally we separate certificate and private key into two separate files and also saved in different directories for better protection.

```
[ec2-user@ip-172-31-84-242 certs]$ sudo systemctl restart httpd
```

A related LAMP installation session was captured and saved in
http://ciast.uccs.edu/coursera/pub/LAMPInstallationSession.pdf
for your reference.

5. Create Project Web Page and Verify Access

The AMI Linux instance is installed with Apache web server and the default web site is located in /var/www/html.

Task 1. Create default web page.

For this project and to test the access control of web access to instances, we like to uniquely identify the web server by creating a default web page with the following simple content as /var/www/html/index.html.

<h1>This is the Apache web server created by <your email address> for CS5910 Coursera Specialization</h1>

where <your email address> is the one you used for your Coursera account. Note that you have control over the access to this web server. Only you and your peer reviewers will have accessed to this web page.

Verify Web Access

Type https://<your instance IP address>/ in the Firefox browser of your local machine to verify if the default web page is up. Here replace the <your instance IP address> with the IP address of the instance you set up. In my case, it is.
Capture the browser image of your default web page as myWebSite.png similar to Figure 51 below.

Deliverable of Project 1a: Submit the captured Firefox browser image of the default web site of the instance as deliverable for Project 1a.

Note that the web browser will warn the web access is not secure, since the certificate presented by the web server is self-signed, not signed by a public Certificate Authority (CA). On firefox browser, click "Advanced" and then "Add Exception…" followed by "Confirmed Security Exception".

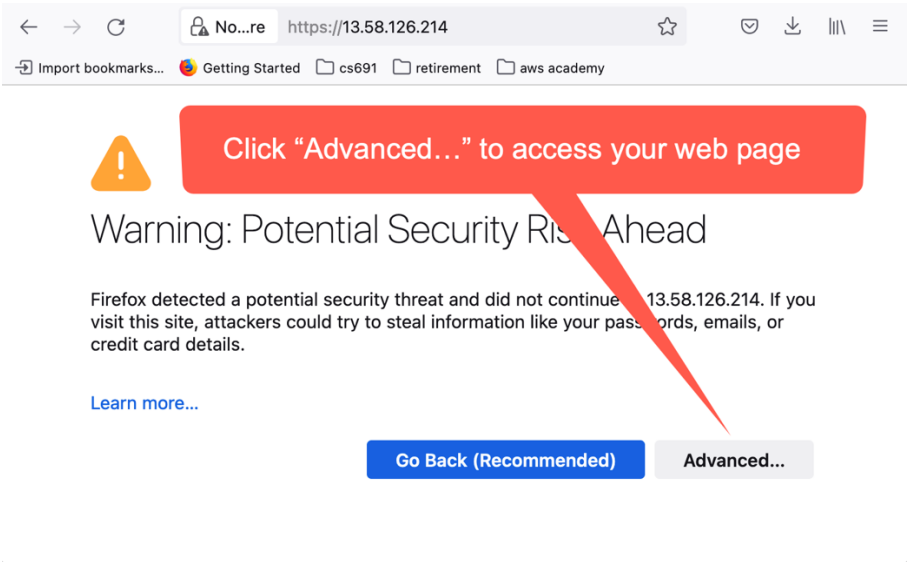

Figure 49. Brower warns the potential man-in-the middle attack.

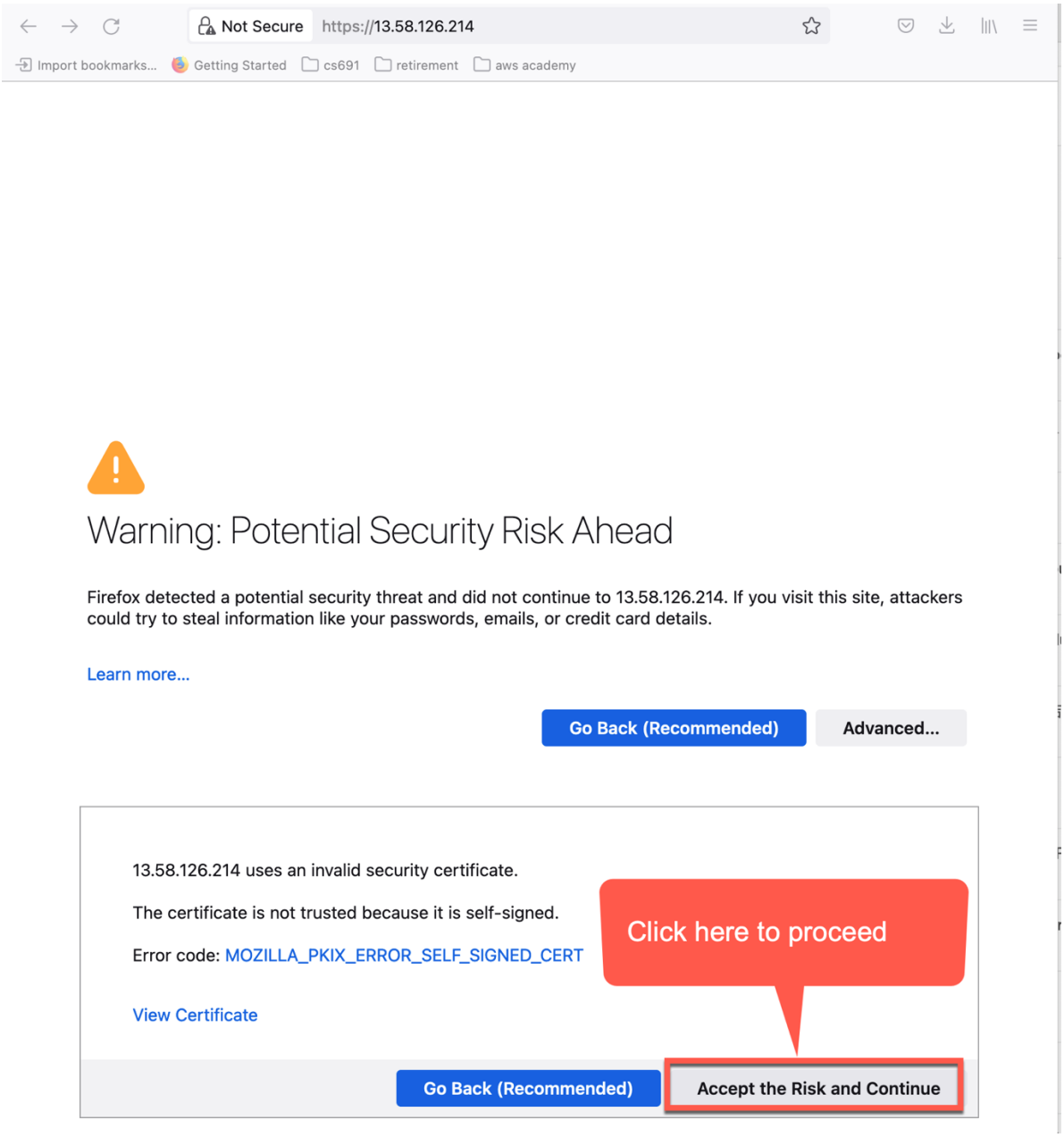

Figure 50. Choose to access the default web page even with warning.

Figure 51. Https access to the default web page with warning.

Make sure you finish your session by clicking the “End Lab”. The results is to shutdown the AWS session. The green icon to the right of “AWS” button will be changed to red color.