

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

For this specialization, I have created an Amazon AWS Image for you to clone an instance and use it for learning the cybersecurity concepts, security policy and related enforcement procedures. To clone an Amazon AWS instance (virtual machine) from the Amazon AWS image, you need an AWS account to login to the management console and use their EC2 GUI control to create the instance.

Your first task is to apply for your free AWS Educate account or a regular AWS regular basic account. Follow the instruction in Section 1. The AWS regular account provides full access to AWS services with all privileges for learning public cloud computing/security. The AWS educate account allows you to work on the creation of AWS EC2 instances for the exercises of this specialization.

Warning: Use your AWS regular account wisely, then you will not be charged for running instances for the exercises in learning our specialization. Make sure you stop the instance each time after you finish your exercise! You may be charge small fees for the storage. Please make sure you set up a billing alarm and follow all the guidelines in the lessons, otherwise you might incur costs. We do not take any responsibility for any costs incurred. See <http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/free-tier-alarms.html> for setting up billing alarm.

In this project, you will learn how to create an AWS account, and clone an AWS EC2 instance in a region closest to you. You will set up a default project web page and a web page that documents the ping results you used to choose the region. You will learn **the availability support by the AWS EC2 service** and how to **restrict access to the web service** of the instance only to you at home by specifying the sources that allow HTTP/HTTPS access with Security Group Interface. You will also learn how the private key is used to access the AWS Linux instance without needing to provide login or password.

1. Create AWS Account

1.1 Option 1. Create free AWS Educate Account with Basic Support.

For this specialization and certificate, we have worked with Coursera and AWS Educate to provide an AWS Educate Starter Account (ESA) with free credit for you to conduct the exercises in this specialization. You will not need credit card for the payment. Follow the instructions in following "AWSSignupGuide_Student" pdf file to request the AWS Educate Starter Account.

http://ciast.uccs.edu/coursera/pub/awseducate/AWSSignupGuide_Student.pdf

Additional AWS Educate Getting Start Resources can be found in <https://awseducate.instructure.com/courses/128>

Your AWS Educate Starter Account will be renewed on an annual basis as long as you are an active student of an education institution and a member of AWS Educate.

If you use the account wisely by stopping the instance after every session, you should be able to finish with all the exercises with the provide free credits.

Note that the non-educate email account such as gmail.com or yahoo.com may not work since AWS Educate may check on the domain name to see if it contains .edu.

1.2 Option 2. Create AWS Regular Account with Basic Support.

Follow the link, <http://ciast.uccs.edu/coursera/pub/CreateAWSBasicAccount.pdf> to register for AWS regular account. This basic account will give privileges to use/learn other AWS cloud services such as Route53 Advanced DNS server for creating your own domain names.

2. Create AMI instance for CS5910 exercises.

Once you setup an AWS account, use the management console to create an AWS EC2 instance following the instructions in <http://ciast.uccs.edu/coursera/pub/CloneCS5910AWSEC2Instance.pdf>

In the creating tag step, please create a tag with key as Name and value as cs5910_<yourEmailAddress>_AMILinux_i1 replacing <yourEmailAddress> in previous value with your email address used in registered Courser account. For example, I use cs5910_cchow@uccs.edu_AMILinux_i1 for the instance name of my EC2 instance. The naming scheme helps keep track of the instances you may have.

In last step of creating the AWS EC2 instance, make sure you create and download the private key for accessing the instance. Please name it with cs5910_<yourEmailAddress>_AMILinux_i1.pem. Make sure you back up your private key in a safe place such as a flash drive. You may want to encrypt it.

Once the instance is initialized and running, click it on the dashboard or hit the Instances of the EC2 main window. Find the public IP address assigned to this instance. We will reference this as <InstancePublicIPAddress> to be used in setting up the access.

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 cs5910_<yourEmailAddress>_AMILinux_i1.pem ec2-user@<InstancePublicIPAddress>
```

Here the parameter right after -i option is the private key file name. Make sure it can only be accessed by you. You can use "chmod go-rw cs5910_<yourEmailAddress>_AMILinux_i1.pem" to remove group and other user access to your private key file.

3.2 For Windows users,

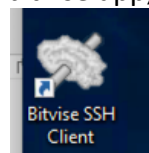
you can download the bitvise SSH Client app installer and install bitvise app. It is available at <https://www.bitvise.com/ssh-client-download>. It provides a nice SFTP gui for drag and drop files between remote server and your local client. It also does not require to go through the .pem to .ppk file conversion which is required by putty app. The putty is a popular command line interface app and can be downloaded at

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

(choose the proper msi file). I recommend using bitvise for accessing our instance. It also saves your configuration time.

3.2.1 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.

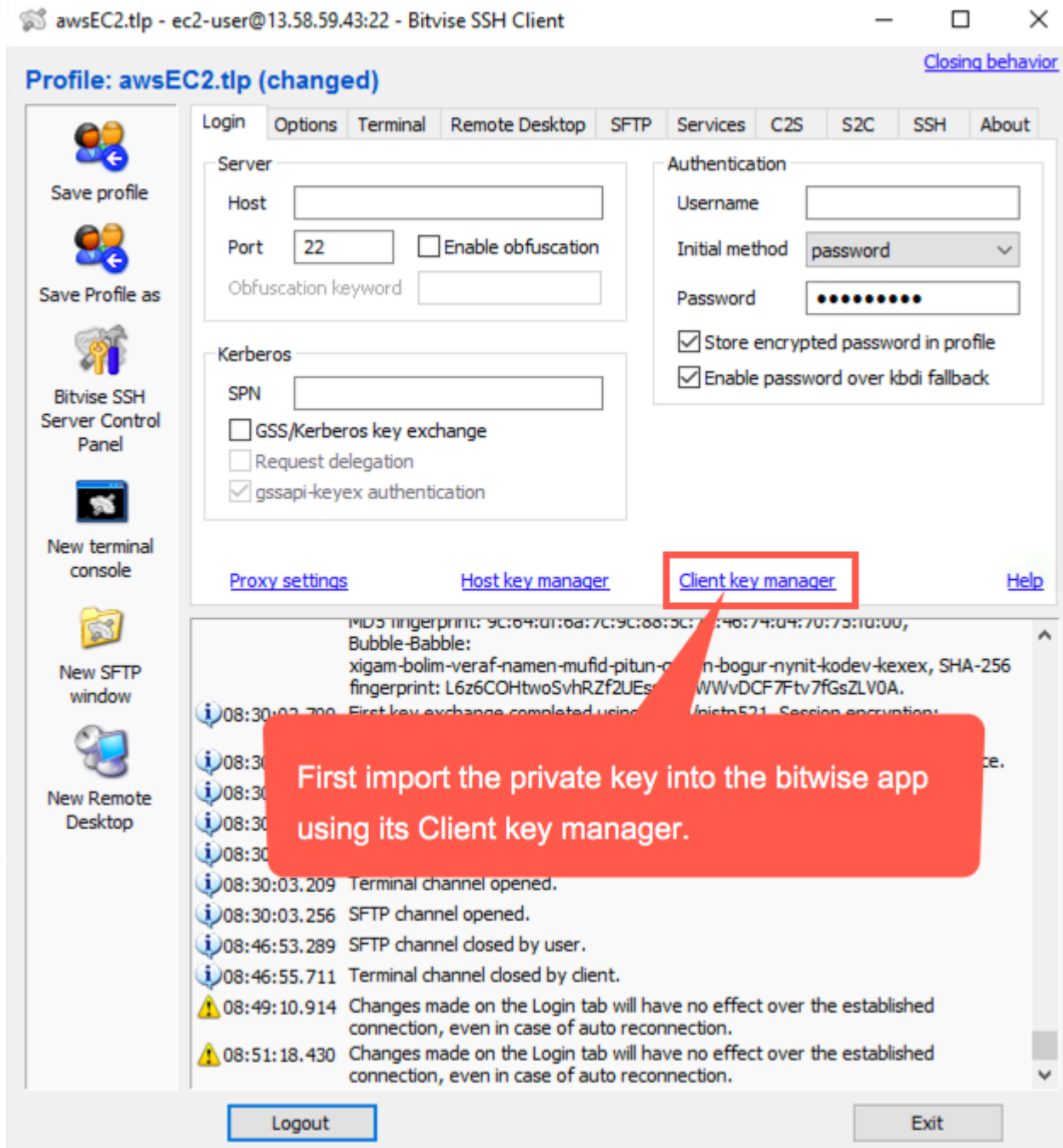


Figure 1. First import the private key into the bitwise app using its Client key manager.
Step 2. Click Import button.

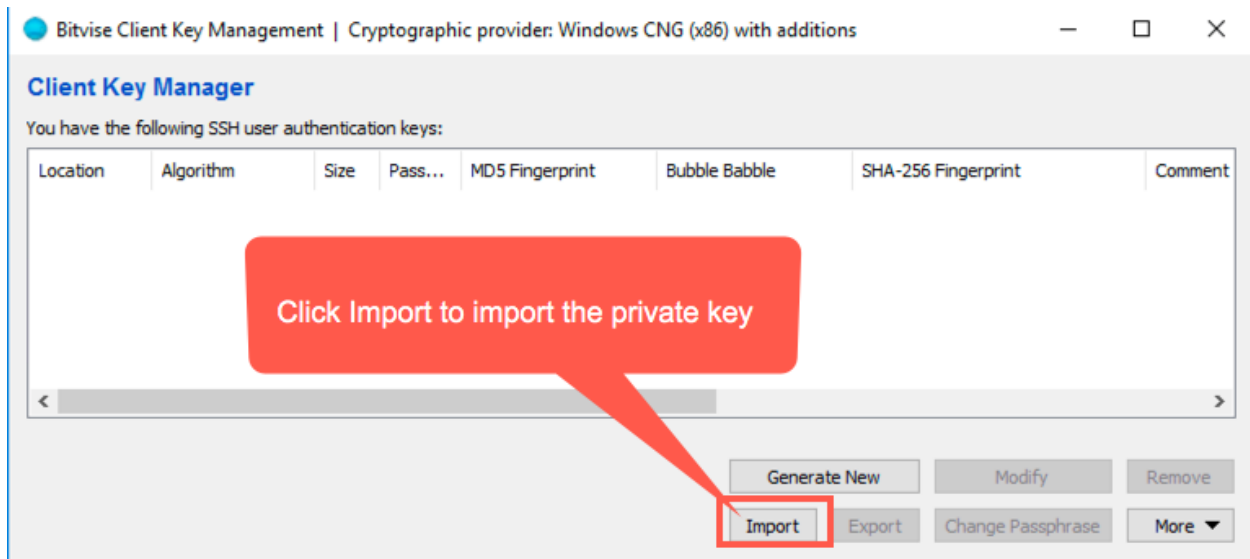


Figure 2. Click Import button to import the private key.

Step 3. Change the file type to all file (*.*) in order to reveal our private key in .pem file format.

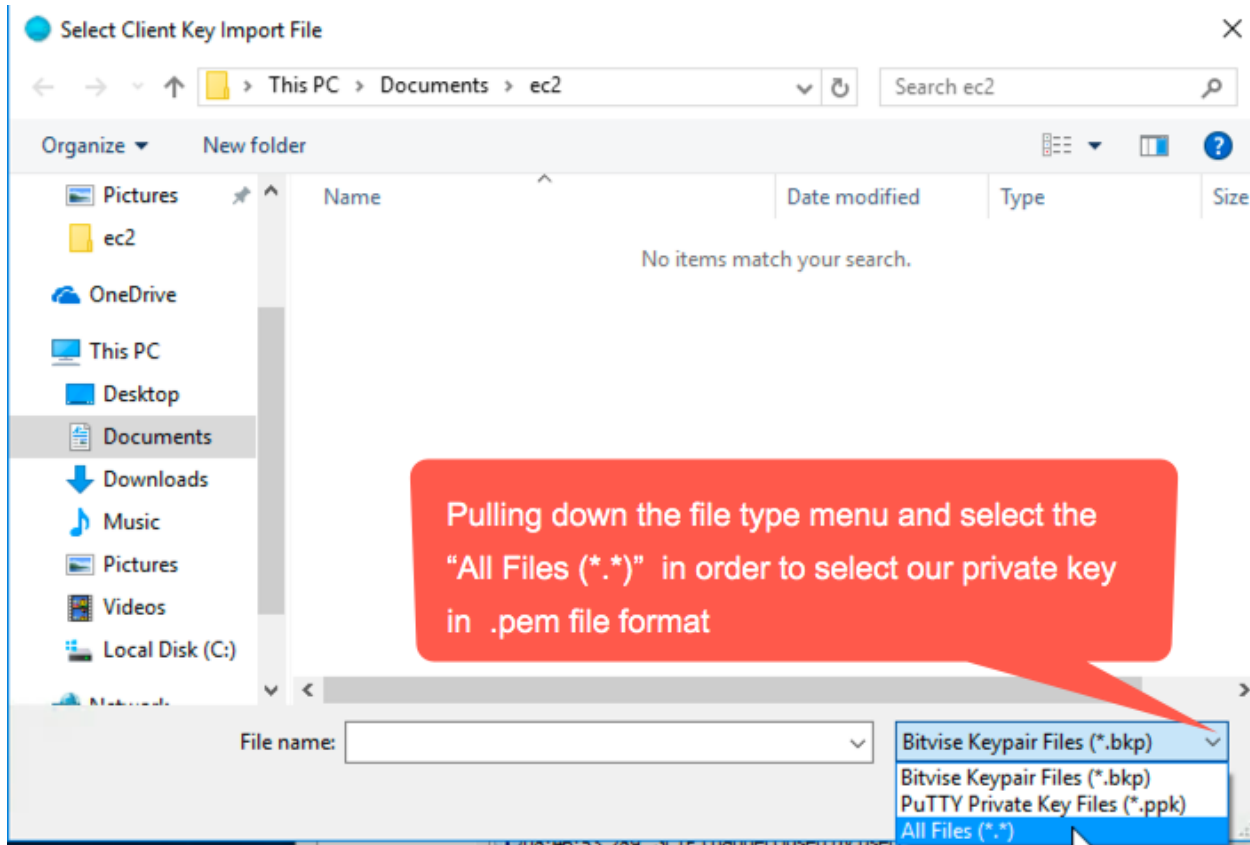


Figure 3. Change file type to All Files (*.*) to reveal our private key in .pem file format.

Step 4. Select our private key and Click Open.

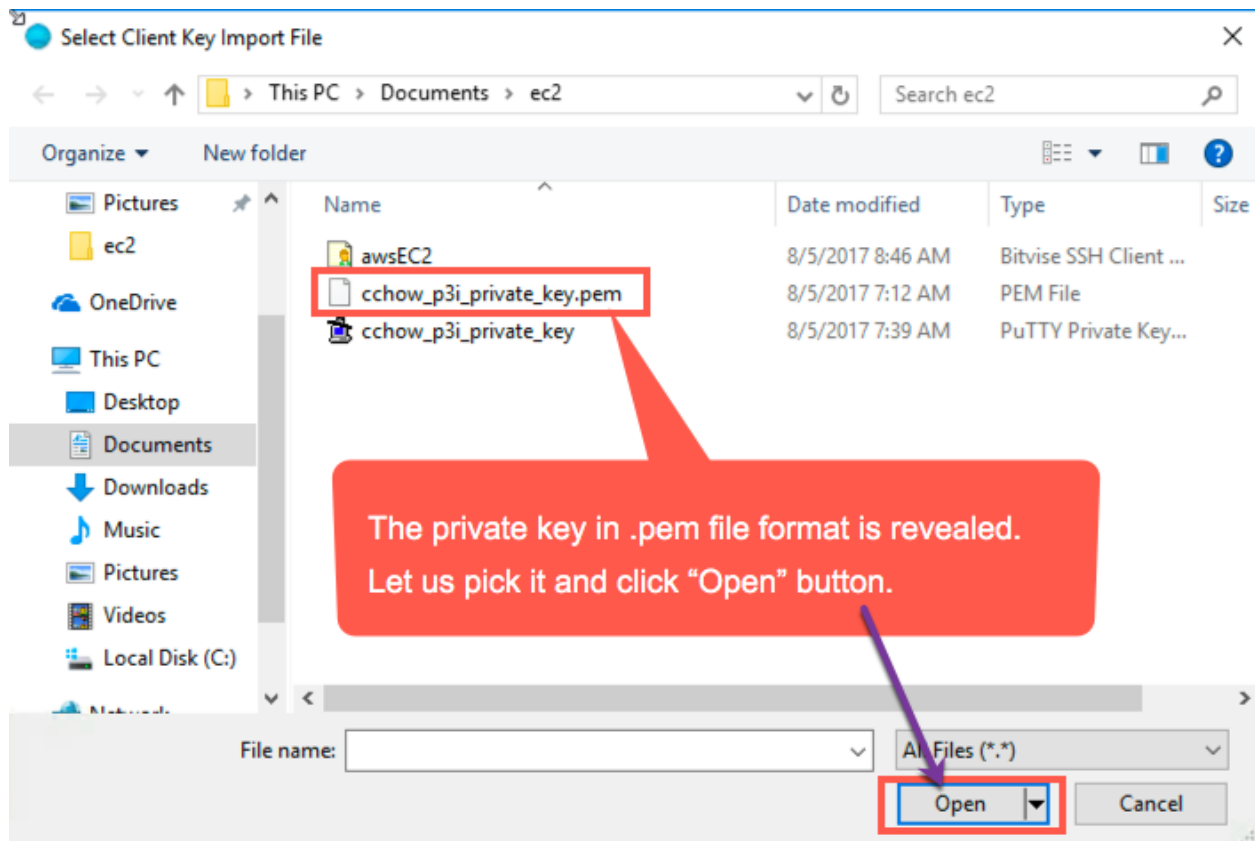


Figure 4. Select private key in .pem file and click Open.

The private key is imported and save in

Step 5. Save it as Global key #1.

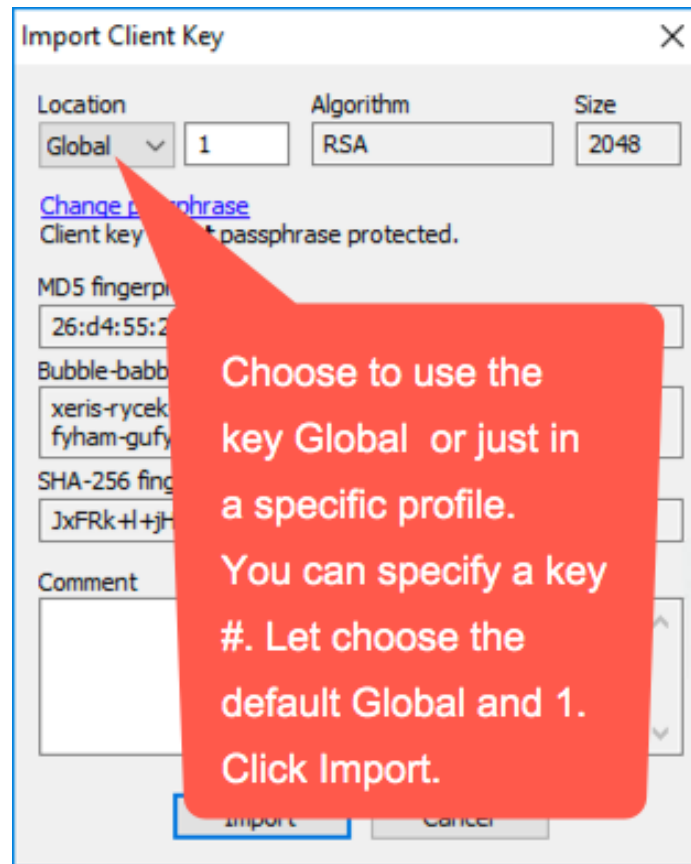


Figure 5. Save it as Global key #1 by default and Click “Import”.

Figure 6 shows the import results.

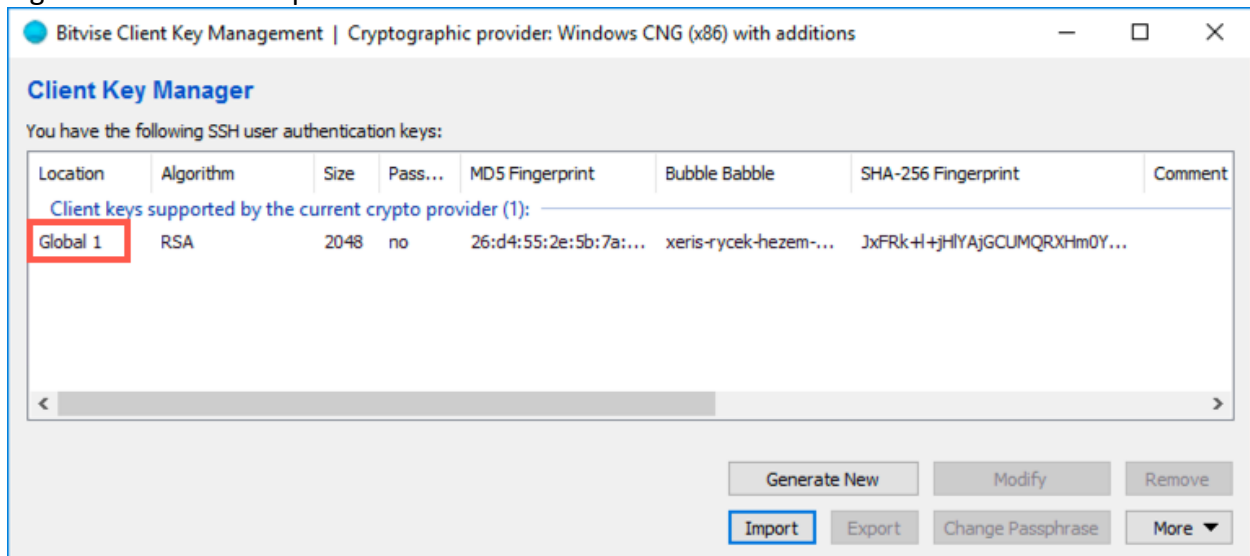


Figure 6. Client Key Manager display the imported Global 1 key, our private key.

Step 6. Specify the instance IP address, username, access method, the private key used for accessing the instance.

Make sure you use “ec2-user” not the “root” for accessing the AWS EC2 instance, since its sshd is configured to not allowed root direct login and there is initially only a single “first” user created. Replacing “password” method with “publickey” method in the dropdown menu of “Initial method”. Verify the “Global 1” key is selected. Click “Save profile” and enter awsEC2 as profile name. Next time we start bitvise, it will remember the current profile. Click Login to verify if it works.

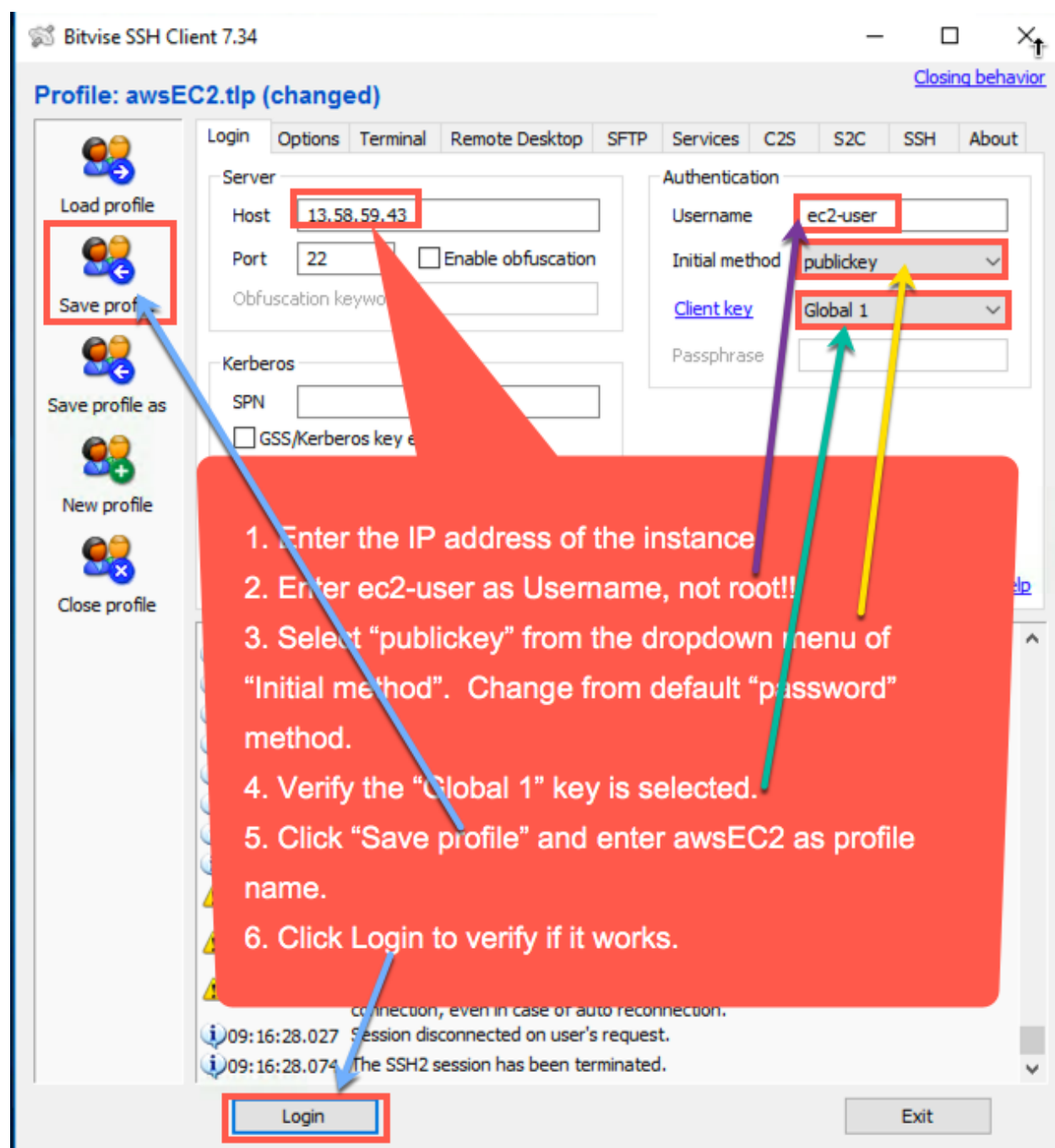
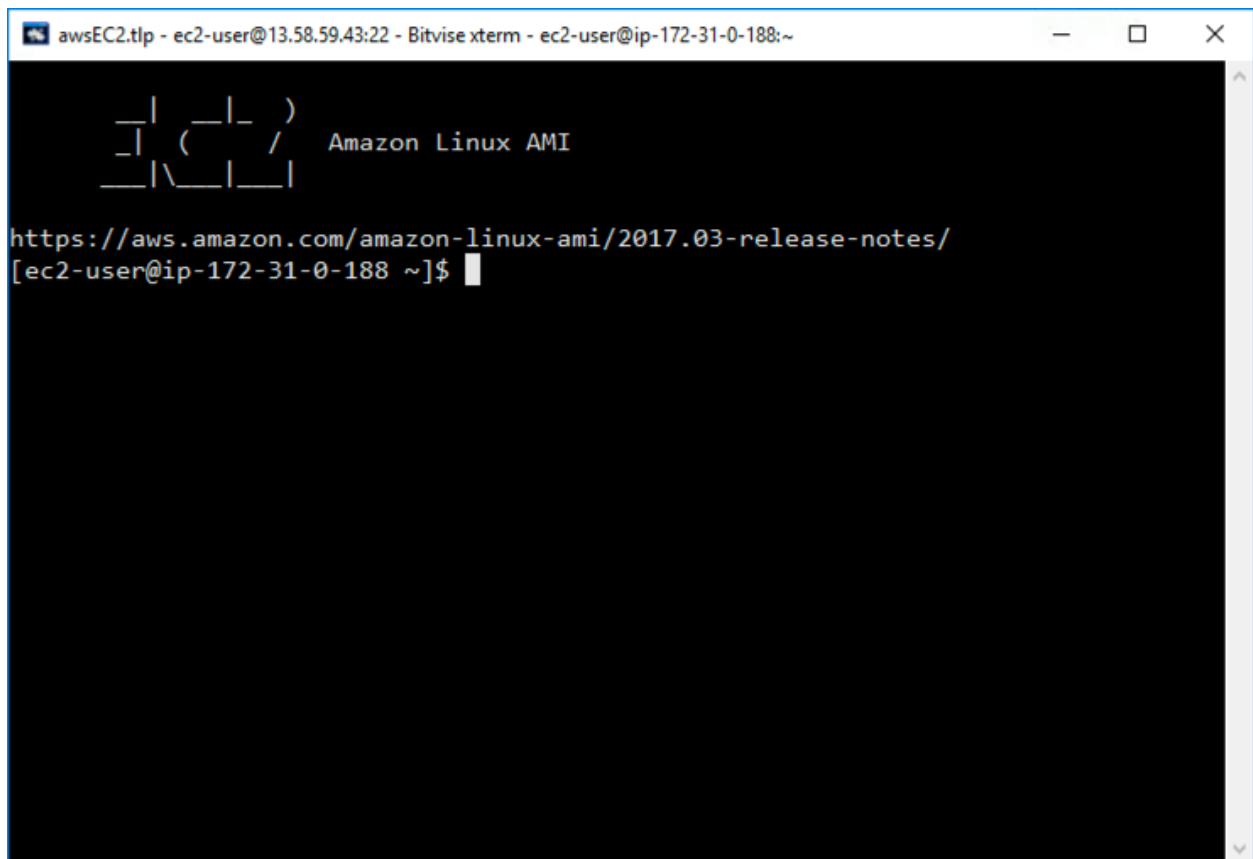


Figure 7. Specify instance IP address, username, access method, the private key used.

Step 7. The bitwise SSH client app and SFTP app are launched.

The SFTP app allows intuitive drag and drop files/directories between the remote instance and the local client.



The screenshot shows a Bitwise xterm window titled "awsEC2.tlp - ec2-user@13.58.59.43:22 - Bitwise xterm - ec2-user@ip-172-31-0-188:~". The terminal displays the Amazon Linux AMI logo, the text "Amazon Linux AMI", and the URL "https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/". The prompt is "[ec2-user@ip-172-31-0-188 ~]\$".

Figure 8. bitwise SSH Client.

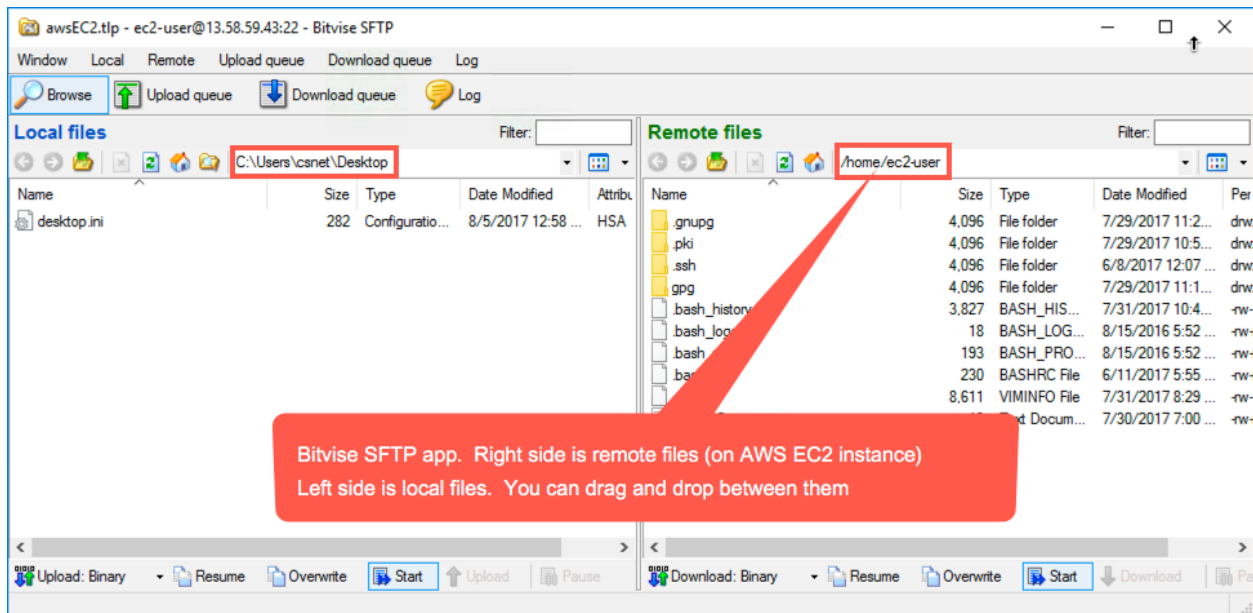


Figure 9. bitvise SFTP app.

3.2.2 Option: Install putty SSH Client

Follow the steps below to setup the private key for accessing your instance using putty.

Step 1. Use PuTTYgen app to convert the .pem file to .ppk private key format

Note that the new putty version does not accept the old .pem file format. Type PuTTYgen or click Windows icon and select the PuTTYgen in the app list just installed.

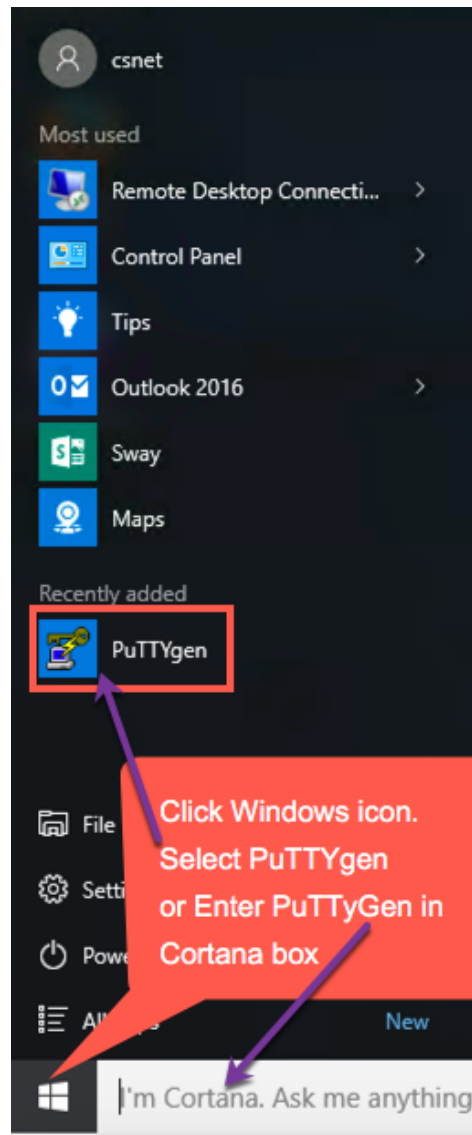


Figure 10. Open PuttyGen app.

Select Import key menuitem in Convert menu to conver the .pem file format to .ppk file format.

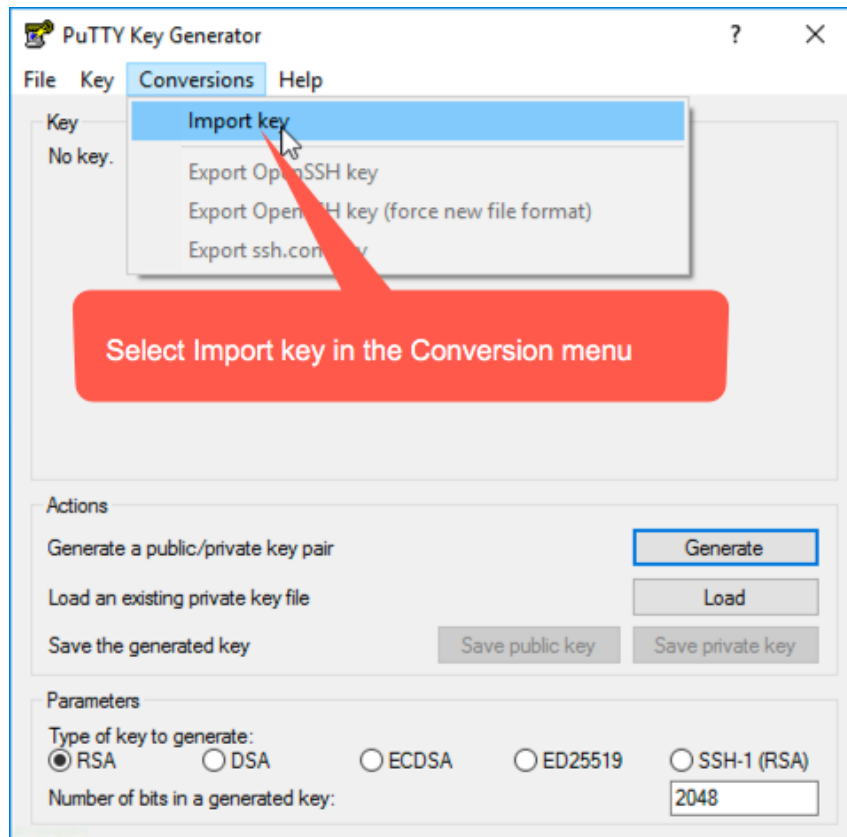


Figure 11. Choose Import key in Convert menu.

After selecting the private key, the app converted the .pem file into .ppk file format. We can click on the “Save Private Key” button to save the file. Make sure you remove .pem in the filename.

Step 2. Start putty app.

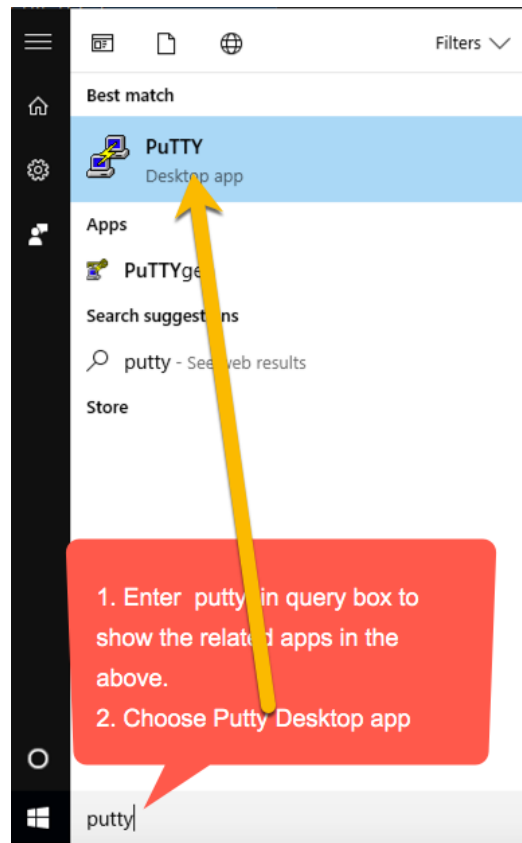


Figure 12. Launch putty app.

Step 3. Enter the public IP address in the Hostname or IP address field.

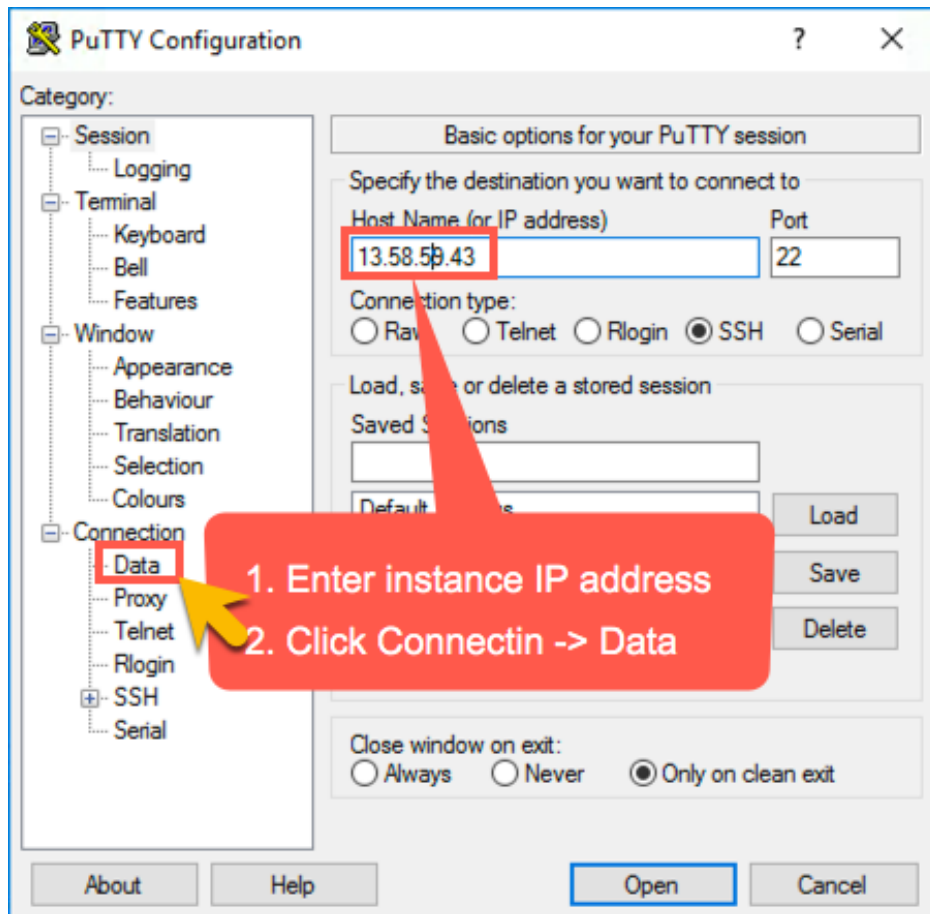


Figure 13. Specify Instance IP address and Clock Connection Data.

Step 4. Click **Connection > Data** in the left-hand navigation pane and set the Auto-login username to **ec2-user**. It is critical we set the username to ec2-user instead of root which shows in my putty set up tutorial.

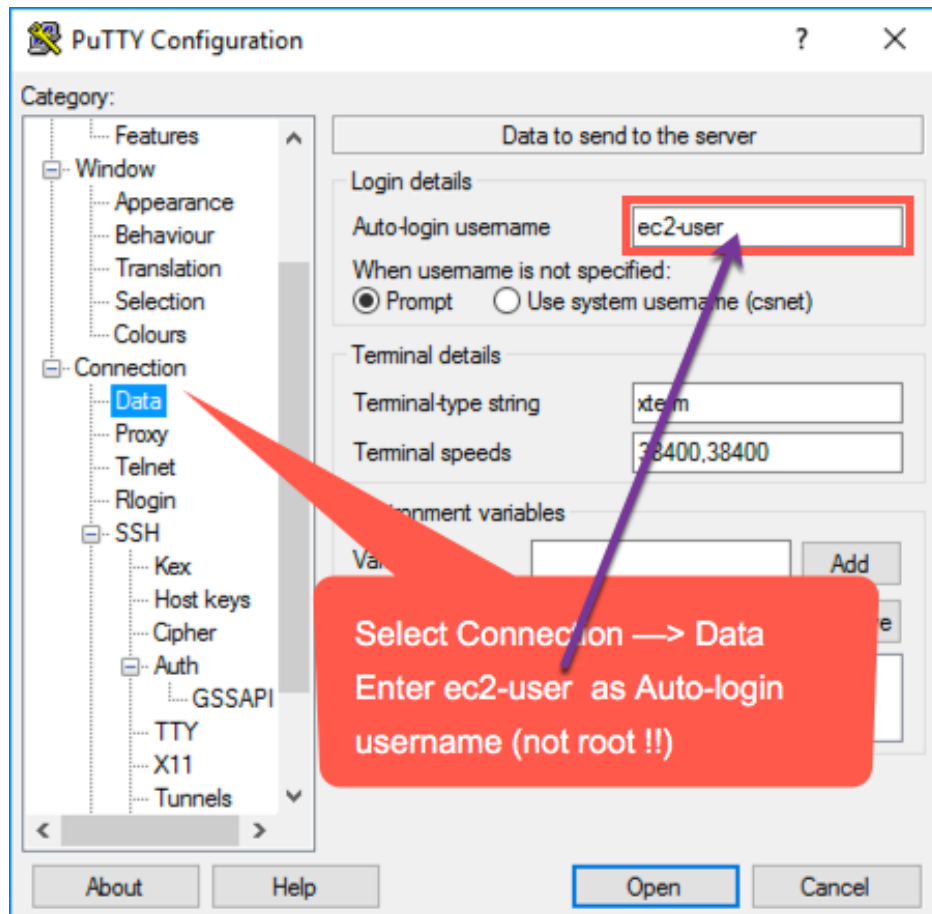


Figure 14. Set Autologin username to root.

Step 5. Click **Connection > SSH > Auth** in the left-hand navigation pane and configure the private key to use by clicking **Browse** under Private key file for authentication.

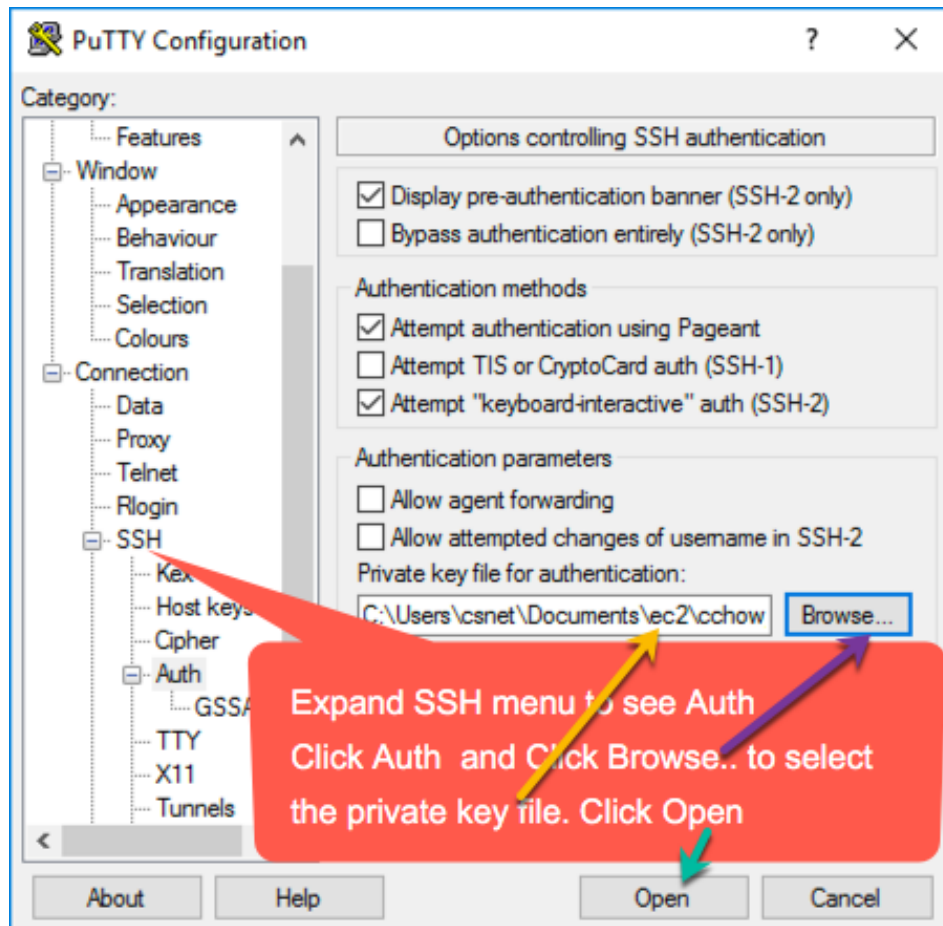


Figure 15. Set private key for the session.

When browsing to select the private key file, make sure you select the private key file in .ppk format which PuTTYgen generates for us.
See Figure 16.

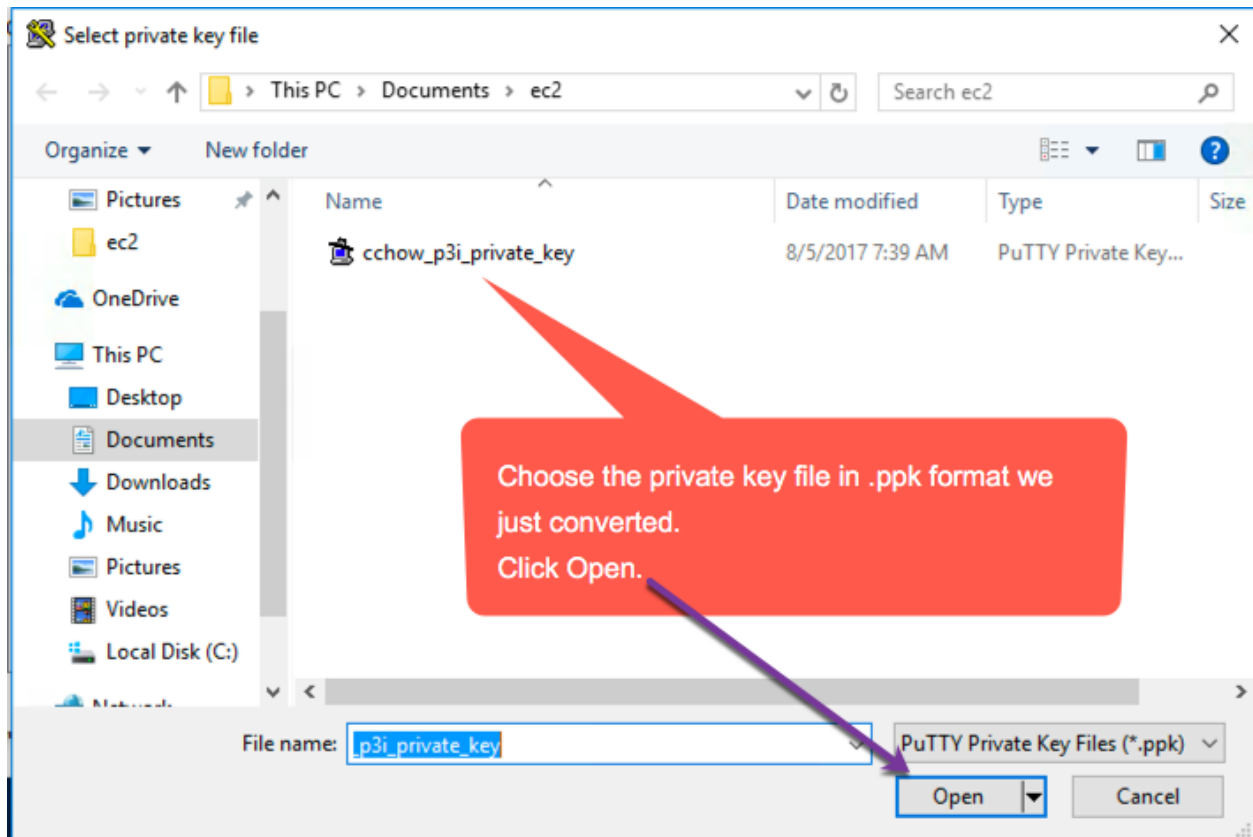


Figure 16. Select the private key file in .ppk file format.

Navigate to the location where you saved your private key earlier, select the file, and click **Open**.

The private key path is now displayed in the **Private key file for authentication** field.

Step 6. Click **Session** in the left-hand navigation pane and click **Save** in the Load, save or delete a stored session section.

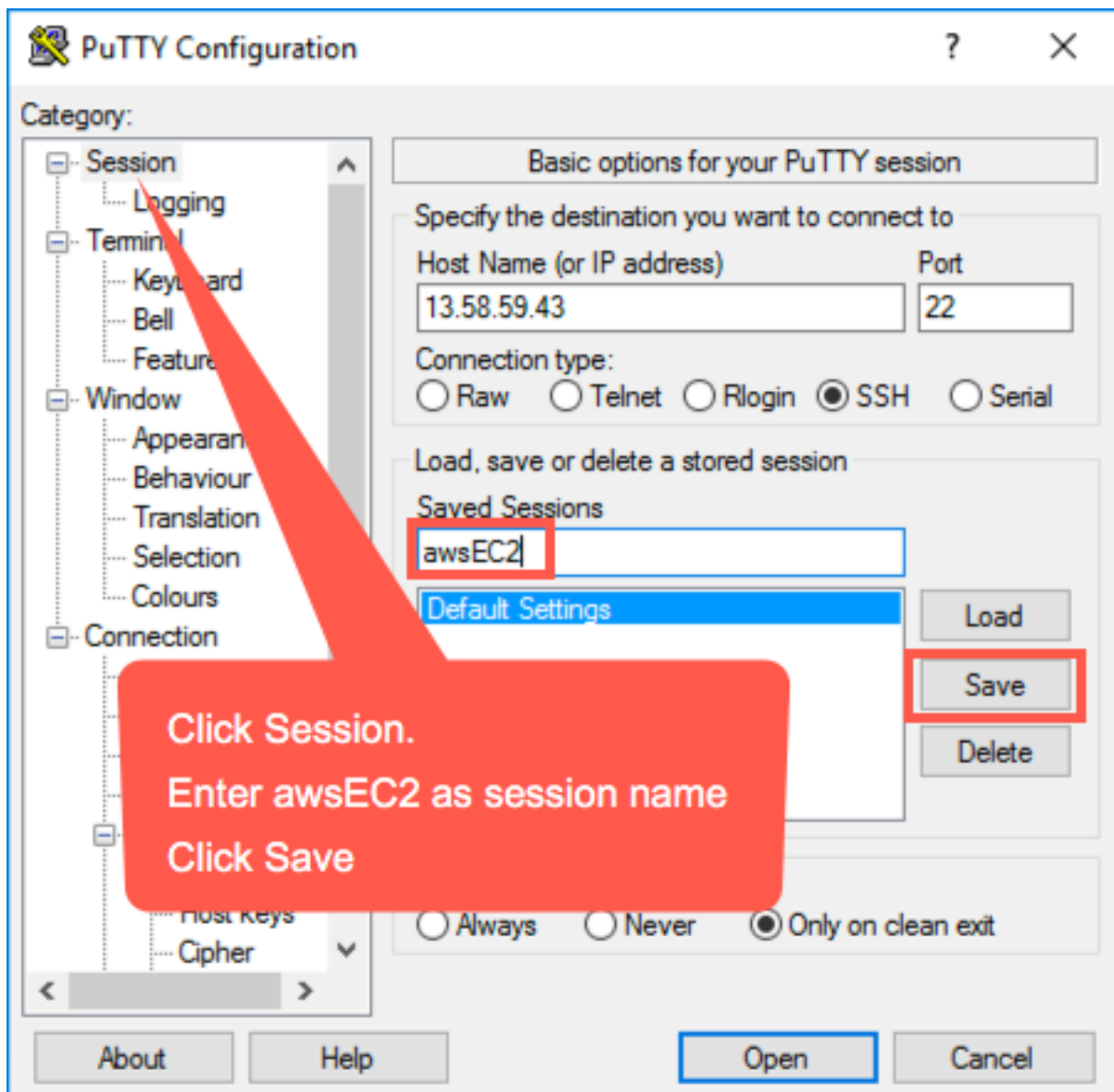
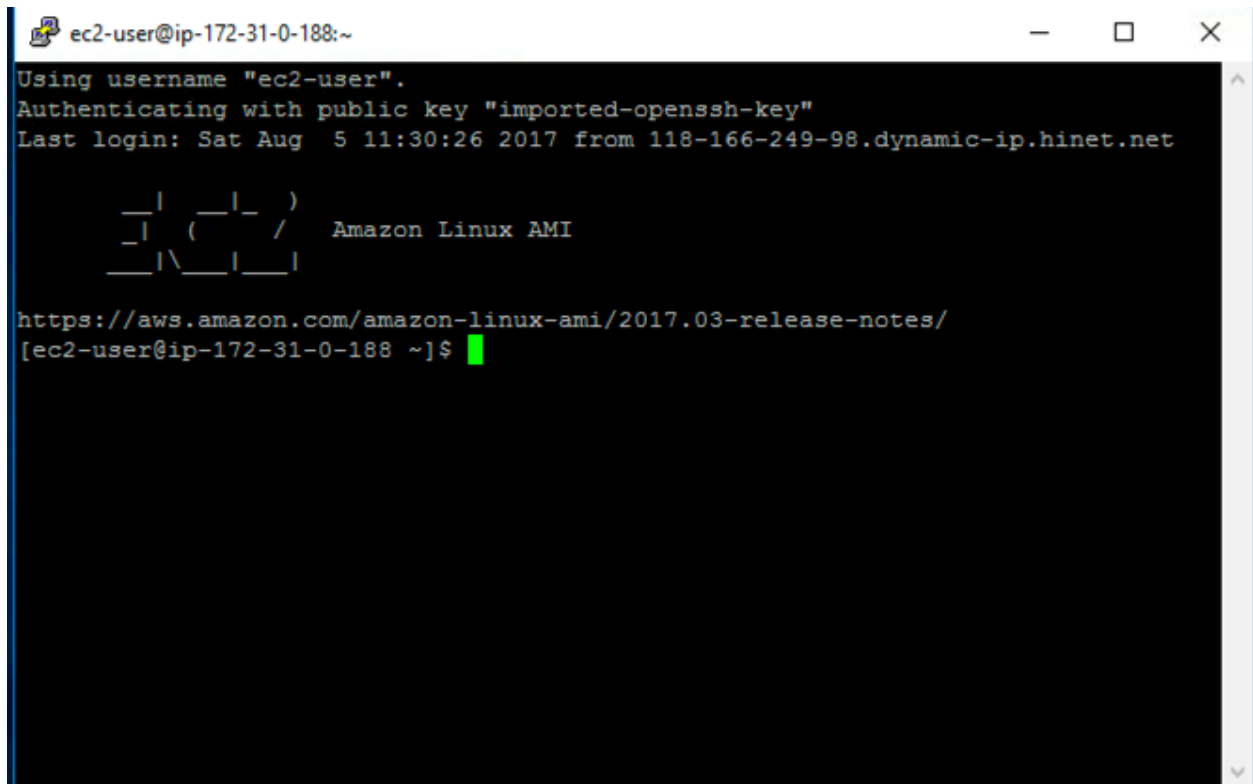


Figure 17. Save Session with awsEC2 as session name for future usage.

Step 7. Click **Open** to begin your session with the server. See Figure 18 for a successful login to our instance.



```
ec2-user@ip-172-31-0-188:~  
Using username "ec2-user".  
Authenticating with public key "imported-openssh-key"  
Last login: Sat Aug 5 11:30:26 2017 from 118-166-249-98.dynamic-ip.hinet.net  
  
  _| _| _| )  
  _| ( _| _| / Amazon Linux AMI  
  _| \ _| _| |  
  
https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/  
[ec2-user@ip-172-31-0-188 ~]$
```

Figure 18. Putty Terminal session to AMI Linux instance without login password.

4. 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.

Task 2. Document how you select the region for this instance and save it as a pdf file in `/var/www/html/project` directory.

The content should be concise, including the ping results of the three regions you initially selected, and the rationale on your decision in selecting this closest region.

Use selectRegion.pdf as file name.

Note that you can create the pdf file on your local machine or laptop, then SFTP it to your web server using bitvise SFTP and type in /var/www/html/project in the right remote directory. Then drag your pdf file from left folder to the right folder.

Verify Web Access

Type http://<your instance IP address>/ in a browser of your local machine to verify the default web page is up. Capture the image of this default web page as myWebSite.png
Type http://<your instance IP address>/project/selectRegion.pdf to see if you can access the pdf file. You should also test https access of these two files. Submit the image and selectRegion.pdf as your deliverable of Project1a.

Note that the web browser will warn the web access is not secure since the certificate presented by the web server is self-signed. On chrome browser, just click “ADVANCED” then “Proceed to 13.58.59.43 (unsafe)” to proceed with access. On firefox browser, click “Advanced” and then “Add Exception...” followed by “Confirmed Security Exception”.

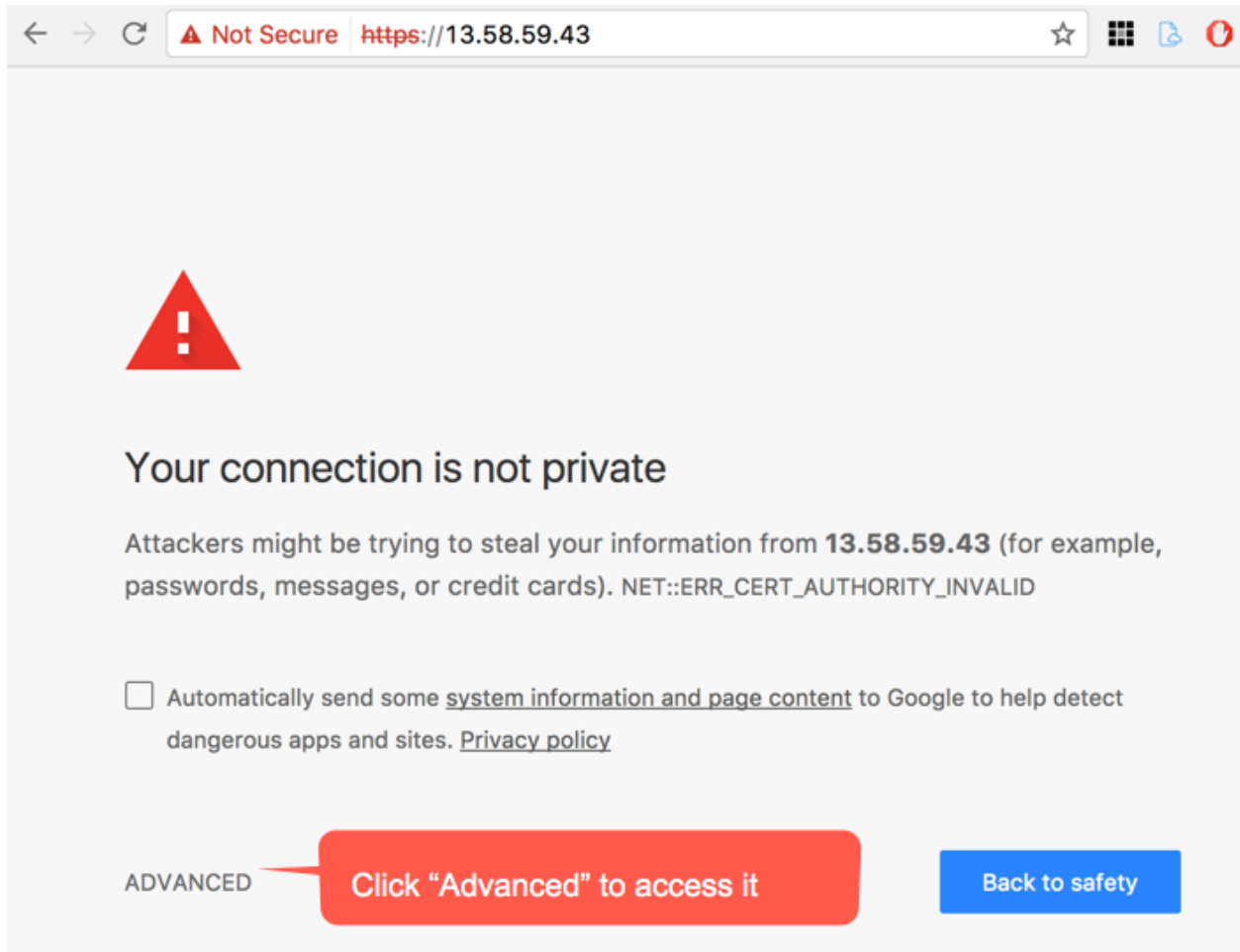


Figure 19. Https access warning.