Practical: Remotely connecting to an EC2 instance

Table of Contents

In this activity you will log in via secure shell to a virtual machine.

Note that there are a number of ways you can connect to the VM - it is not necessary for you to use all of them today. Please talk to your tutor if you run into difficulties or you are unsure of what to do.

Prerequisites and References

- Complete previous exercises and be familiar with accessing AWS.
- AWS: **What is an EC2 (VM)** (https://aws.amazon.com/ec2/)
- AWS: **Guide to Build an EC2** (https://aws.amazon.com/getting-started/launch-a-virtual-machine-B-0/)
- QUT AWS SSO link: **https://d-97671c4bd0.awsapps.com/start#/** (https://d-97671c4bd0.awsapps.com/start#/)

Step 1: Create a Linux virtual machine in the QUT AWS cloud

If you don't already have an EC2 instance running, follow the instructions in **Practical: Creating an EC2 Instance** (https://canvas.qut.edu.au/courses/20367/pages/practical-creating-an-ec2-instance) . After completing this step you will have a machine running, ready for remote

connection and management. We will use Ubuntu 24.04 for this year. You may use later/Older versions if you wish and if you have some Linux expertise.

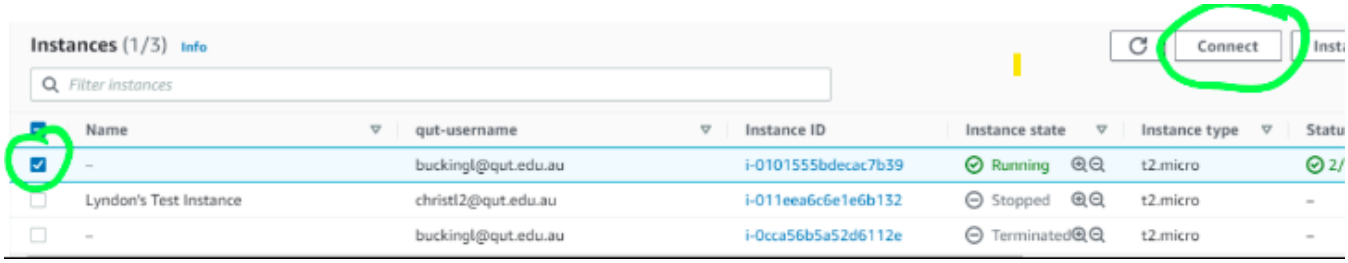
Background: You will use an Amazon t3.micro instance running Ubuntu 24.04 to complete the rest of this activity. Always use the smallest instance that allows you to complete the task. Higher spec machines cost more and you won't often need their capabilities.

Step 2: Connect to your VM with the web-based Session Manager

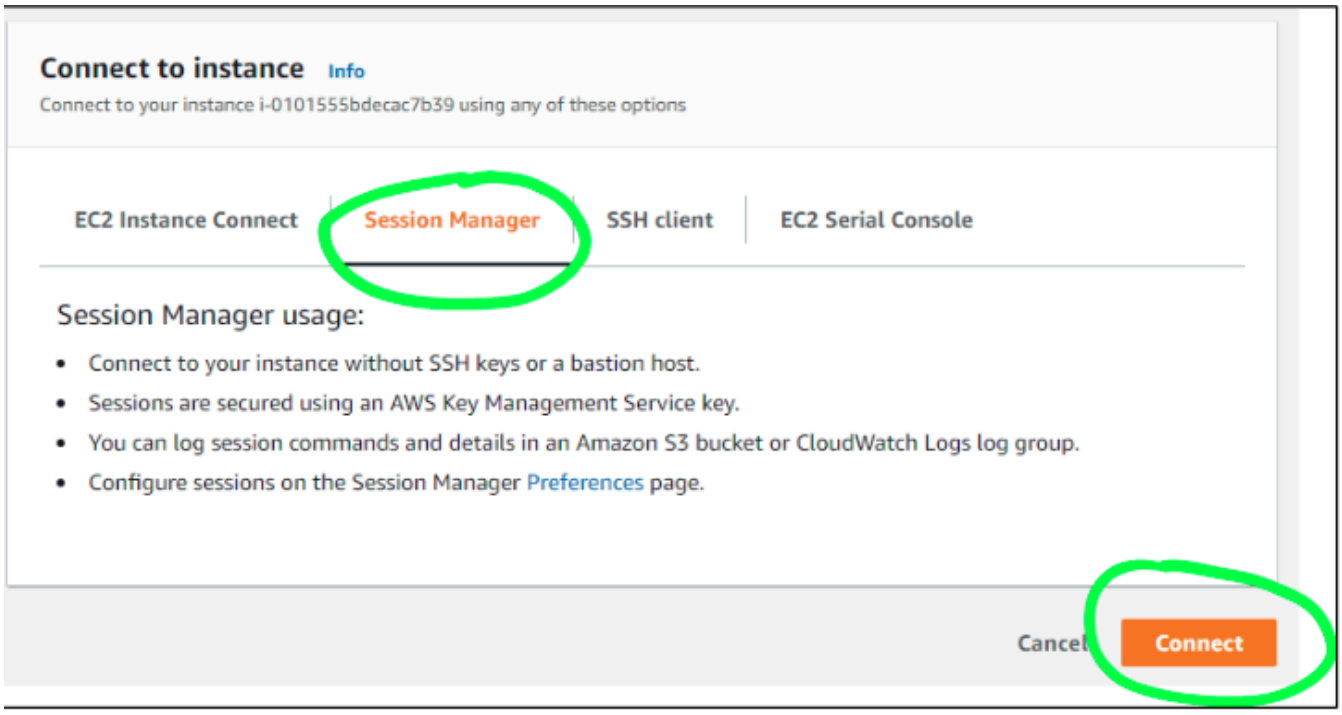
After completing this step, you can work on your VM in a browser tab.

Background: The AWS web-based Session Manager provides convenient interactive access to the VM command line environment.

1. Start your virtual machine and navigate to the AWS Instances page.
2. Select your running instance and click "Connect".



3. Go to the "Session manager" tab, and click "Connect"



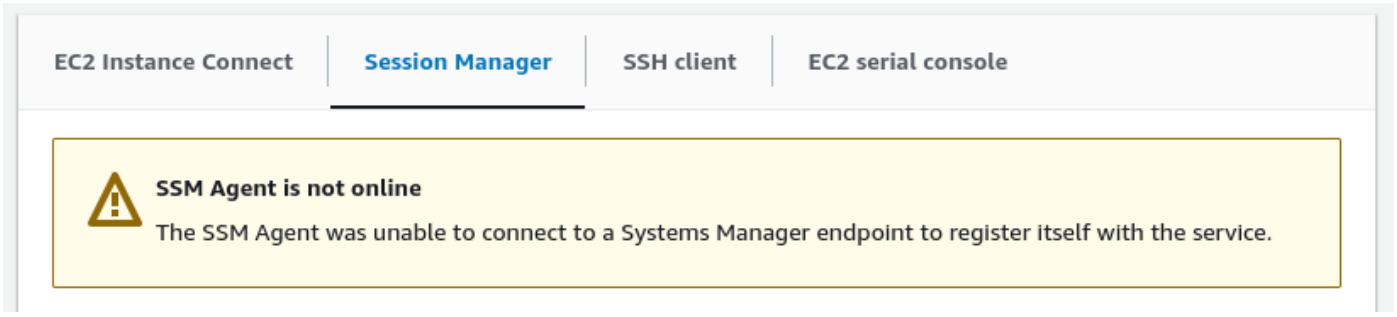
4. Successful log in will lead to a Linux bash session on the VM, hosted in your browser.

5. The Session Manager logs in as the `ssm-user` user. This user does not have the permissions that we want, so you should change to the `ubuntu` user by typing `sudo su -l ubuntu`. This will log you in as the `ubuntu` user and change directory to the `ubuntu` user's home directory.
6. When you are done with the console, you can click *Terminate* in the top right corner. Note that this terminates the *console session*, not the EC2 instance. You'll need to terminate the instance separately if you are done with it.

The Session Manager is handy, but is limited to console access and does not have the additional functionality of SSH, such as file transfer and port forwarding. We recommend that you also explore SSH access.

Common problems:

If you seen an error about the SSM Agent, then the mostly likely problem is a missing IAM role in the EC2 instance, or the security group was not set correctly. You can check these in the instance's details page. If these are not set correctly then it is easiest to terminate the instance and create a new one.



Make sure that you switch to the `ubuntu` user.

Step 3: Get yourself set up with a secure shell (SSH) client

This step describes the pre-requisites needed to connect to your VM via SSH using a command shell or GUI client.

Note: avoid using SSH-remote built-in to VSCode. We have often experienced issues with that because VSCode installs software on the instance to enable advanced features that are not actually needed in this practical. That additional software may cause the instance to run out of memory.

Once you have completed these instructions, Step 5 tells you how to put this into practice to actually connect to the machine. These steps provide an important alternative to the approach in Step 3 above.

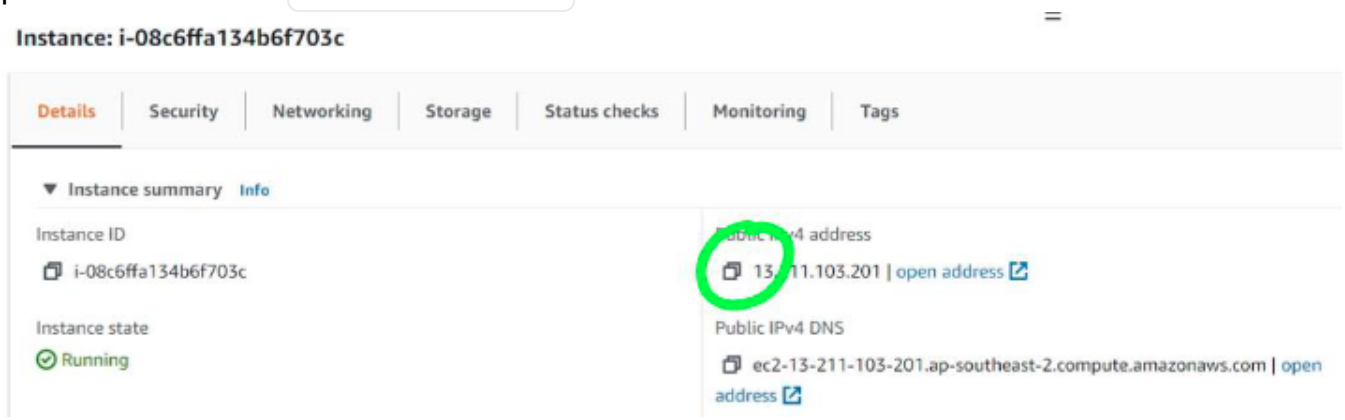
Background: Secure shell (SSH) is the primary management tool for Linux virtual machines. File transfer can be achieved using SFTP or SCP.

- MacOS, Linux (including WSL2):
  - You may find that a SSH client is installed already. To verify that you have the tool, open a terminal window and enter the command: `which ssh`
  - If ssh is present the response will be something along the lines of: `/usr/bin/ssh`
  - If the command is not found, you can use your system's package manager to install an OpenSSH client. In Ubuntu the installation command is: `sudo apt install openssh-client`
- Windows
  - Windows 10 and 11 have `ssh` is already installed. Open the shell (Windows key + R, then type `cmd`) and enter `ssh` at the command prompt to verify that it is installed.
  - Plenty of choices are available, but we have provided software and detailed instructions for some simple and practical options.
    - Modern Windows versions provide the Windows Subsystem for Linux (WSL2). This is a good choice if you would like a reasonably full Linux experience.
    - Git is a source code control system which originated in Linux. The Windows port includes a powerful subset of the Minimal GNU for Windows (MINGW) Unix-like environment.
      - The MINGW ssh and scp clients understand your VM's.pemfile without further processing.
      - The MINGW terminal providesbash, a good script host which will be handy in later learning activities.
    - Putty is a dedicated terminal emulation program for Windows which includes separate SFTP and SCP clients. Putty does not understand .pem files, so you'll have to use the puttygen program to convert your.pemfile to a.ppkfile for use with Putty.

Step 4: Use a secure shell (SSH) to connect to the running VM

Using the VM's private access key with your SSH client allows you to log on to the VM, install software, and execute programs.

1. Start a VM via the AWS management console, and make sure you have access to the `.pem` private keyfile.
2. To be concrete, in this walk through we use a key file which has been saved locally as `Q:\CAB432\2021_02\AWS_Keys\LawrenceCAB432_key2.pem`
3. Go the EC2 instances page and copy the public IP of your instance. In this walkthrough, the public IP address is `13.211.103.201`



. You can also use the Public DNS instead of the IP address.

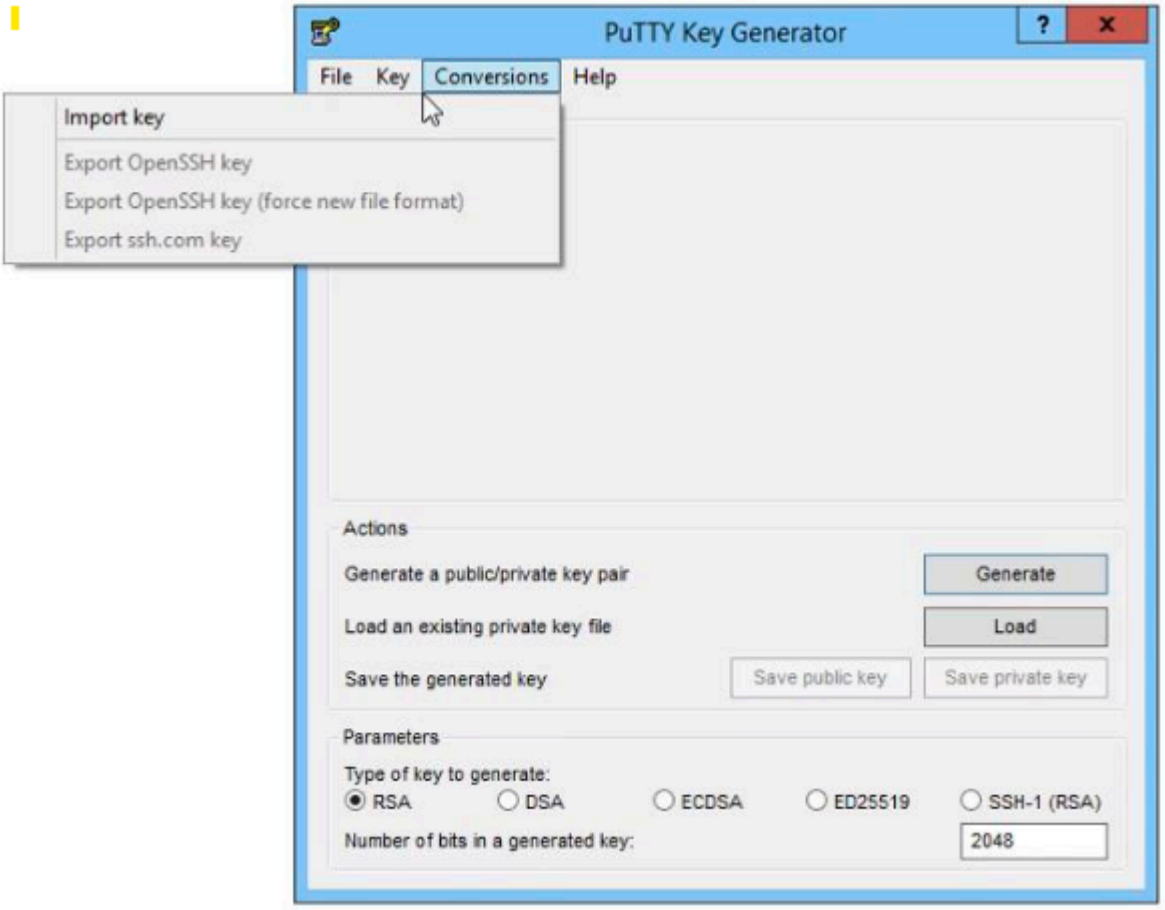
4. **Please note! QUT VPN access is currently broken and being investigated** If you are not on the QUT campus, then you will need to connect to the **QUT VPN (https://qutvirtual4.qut.edu.au/group/student/it-and-printing/wi-fi-and-internet-access/accessing-resources-off-campus)** . The security policies do not allow SSH access from arbitrary non-QUT IP addresses. Talk to the teaching team if using the QUT VPN is not feasible for you.

It is also possible to get ssh access via the AWS CLI. Check out **AWS Command Line Interface (https://canvas.qut.edu.au/courses/20367/pages/aws-command-line-interface)** and **Using ssh with EC2 for login and transferring files (https://canvas.qut.edu.au/courses/20367/pages/using-ssh-with-ec2-for-login-and-transferring-files)** .

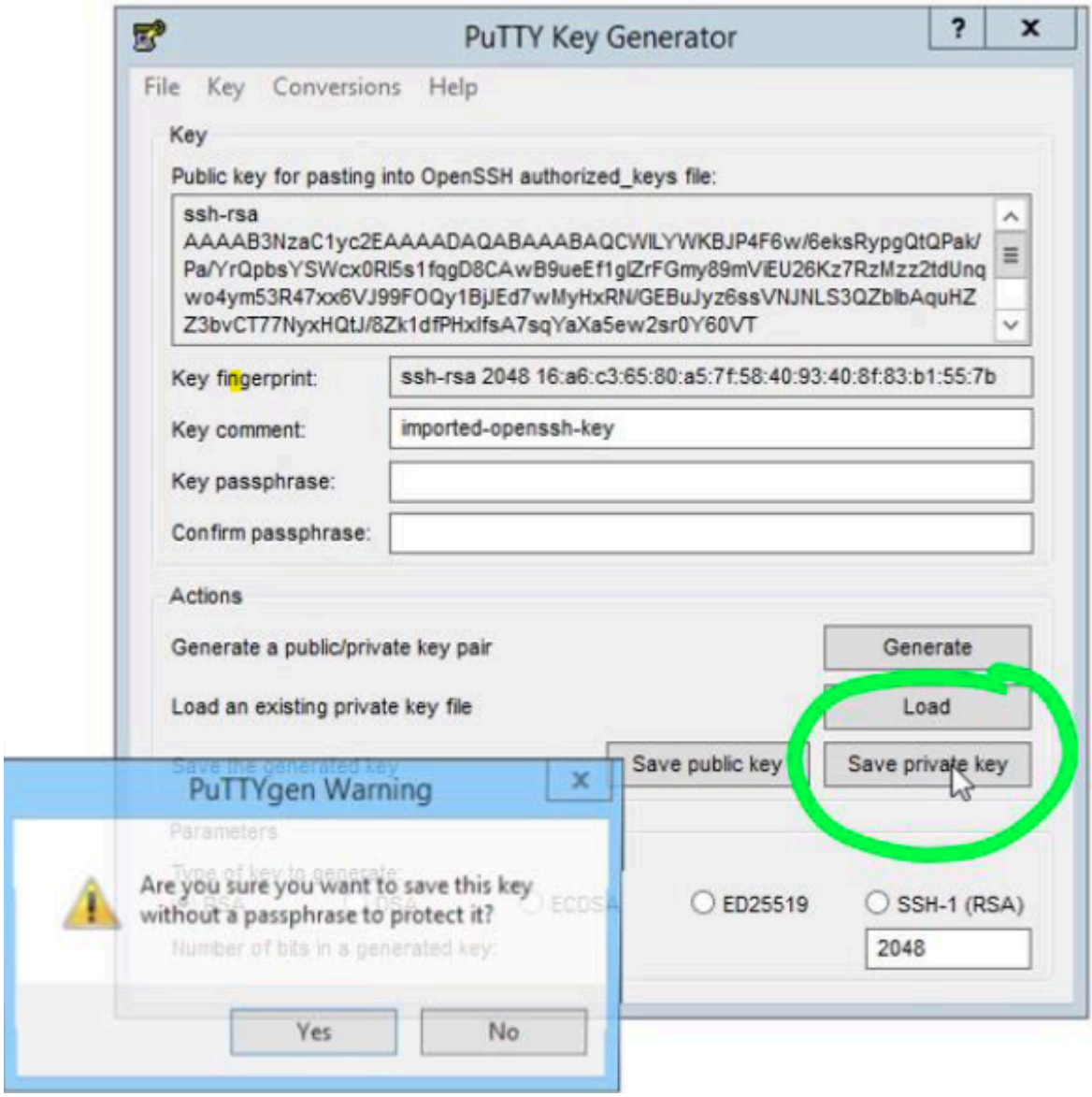
Step 4.1: Use a secure shell (SSH) client (Putty) to connect to the running VM

You can skip this step if you are using a command-line ssh client. Mac and Linux users will not be able to use Putty.

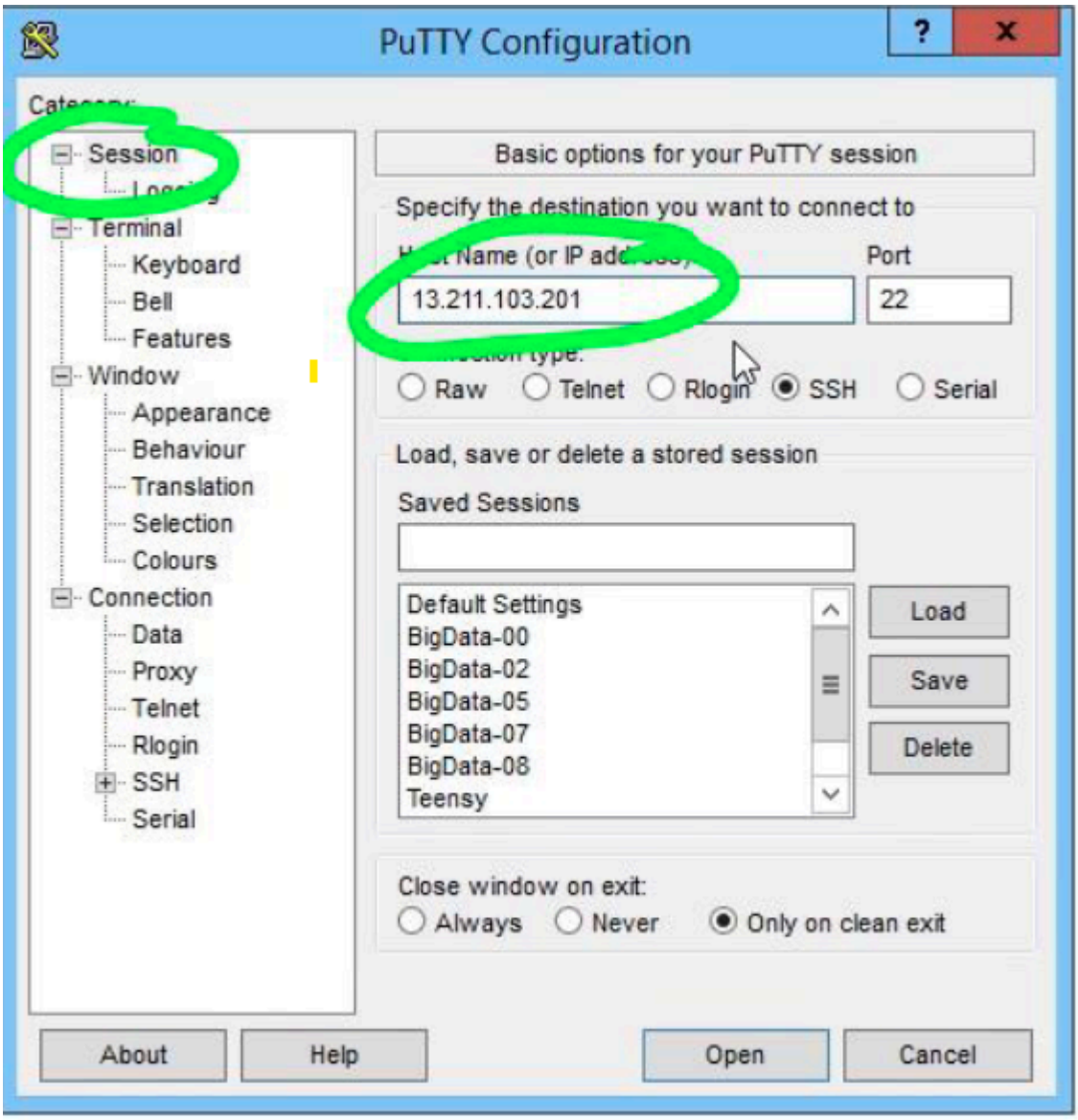
1. Ensure you have putty installed.
2. Run puttygen to convert the OpenSSH key file to a Putty key file
3. Import your `.pem` private key file



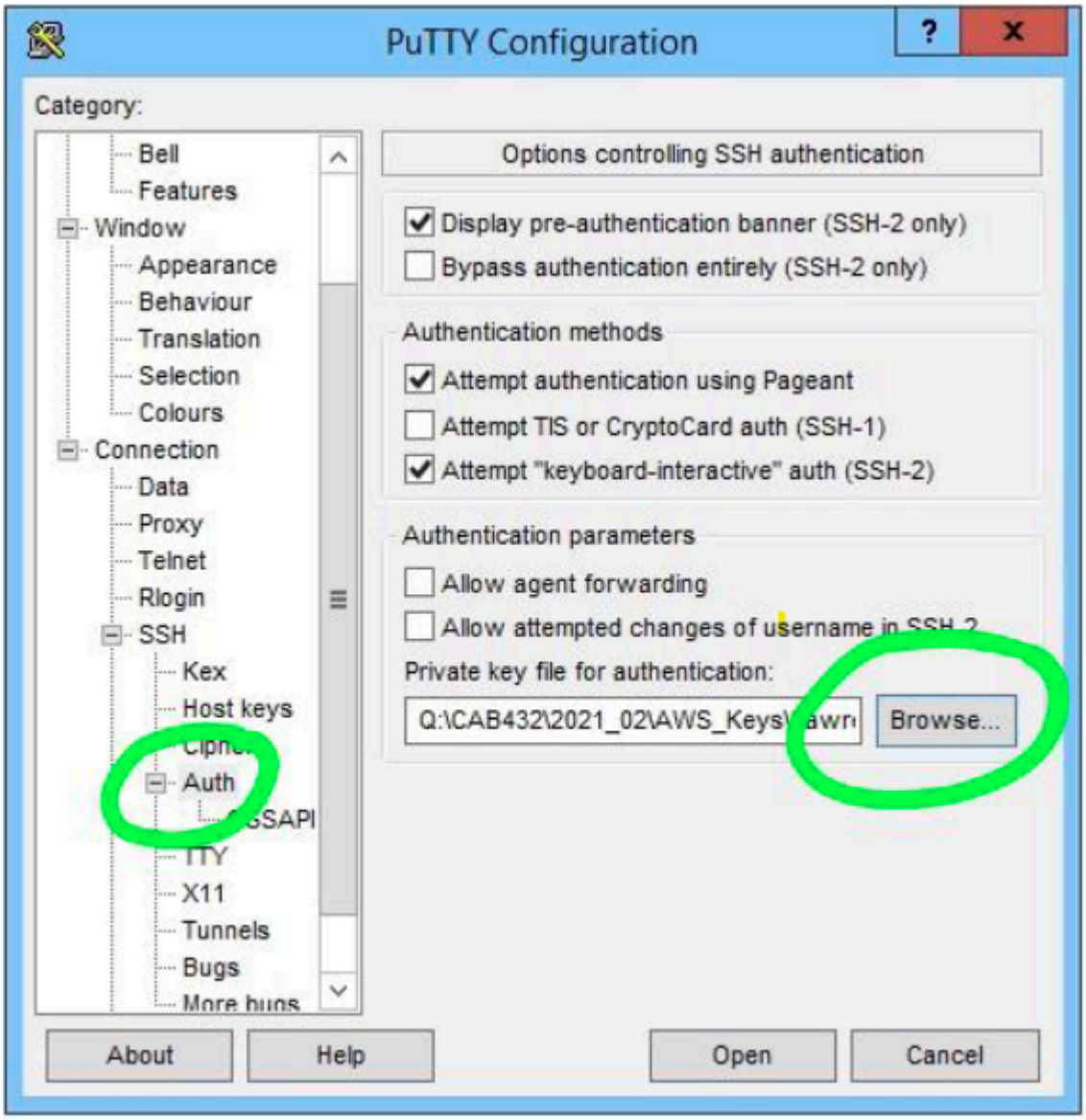
4. Save the private key file. In this example, we saved as `Q:\CAB432\2021_02\AWS_Keys\LawrenceCAB432_key2.ppk`



5. Run putty, and on the session tab, insert public IP address of VM



6. On the connections tab, open the private key (.ppk) file.

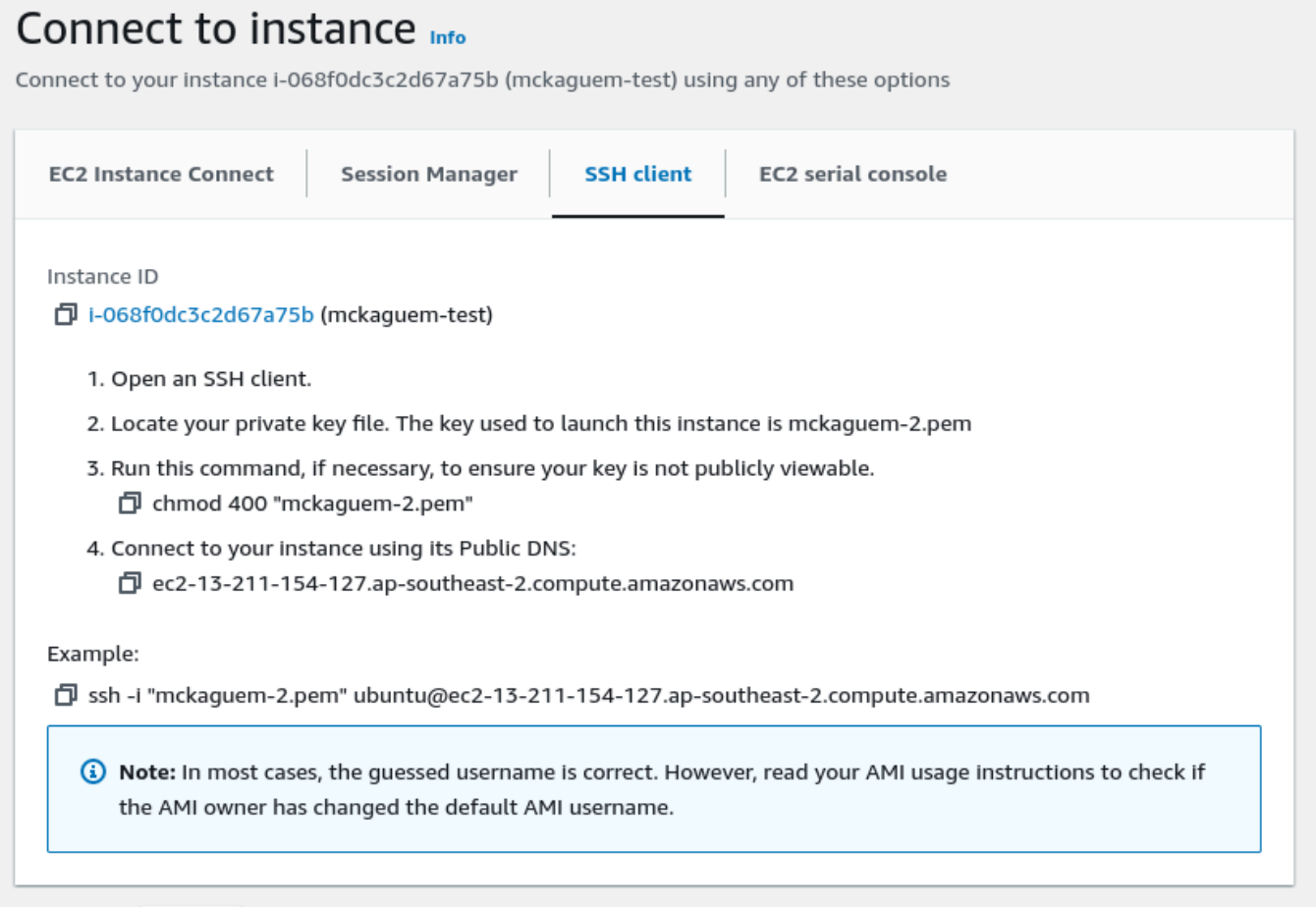


7. You are now set. Open the connection, accept the prompts if any.
8. Login as the user `ubuntu` when prompted. This will open a linux bash session on the VM.

Step 4.2: Use a secure shell (SSH) client (via the command line) to connect to the running VM

1. Make sure you have a suitable command line environment set up, e.g., Linux, MacOS, WSL, or the Windows command line (from Windows 10).
2. Open a terminal window with access to ssh
3. Change directory to where the saved private key file is located (created when launching the instance)
4. As with the session manager method above, navigate to the *Connect to instance* page. Click on the *SSH Client* tab. This will give you instructions on how to connect, including a full

command with all required information.



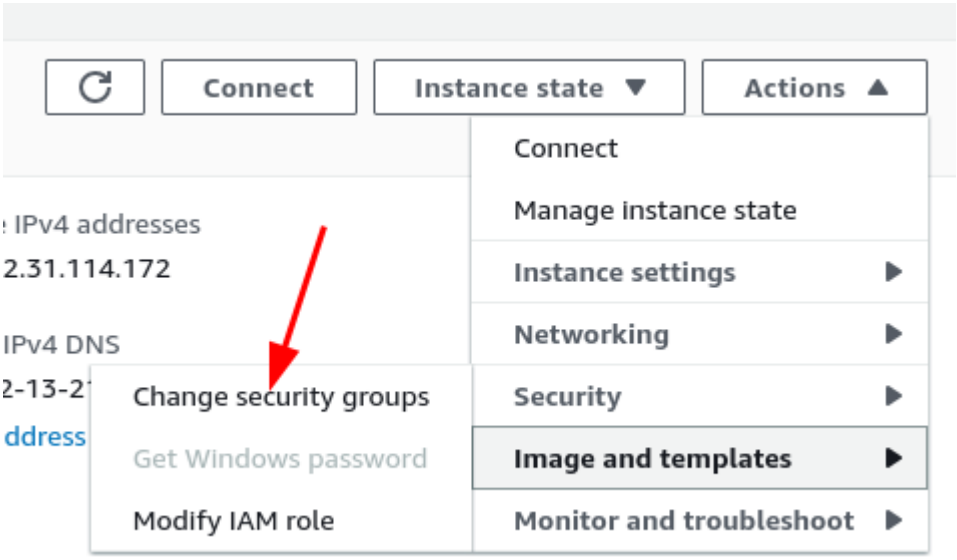
5. Copy the `chmod` command (step 3. on the SSH Client tab) and paste into your terminal. This command is not necessary for Windows users (unless you are using WSL). Hint, click the small square to the right of the command to copy the command into the clipboard.
6. Copy the `ssh` command, labelled "Example" at the bottom of the SSH Client tab, and paste into your terminal.
7. When prompted to save the key, enter `yes`. Successful log in will lead to a Linux bash session on the VM.

Common problems

If the `ssh` command hangs for a long time then networking is likely a problem. On the EC2 instance details page, double check:

- You are on campus or connected via the QUT VPN
- The instance is using a public subnet
- The instance has a public IPv4 address
- On the *Security* tab, check that the security groups are set correctly. You should have `CAB432SG` as the only security group.

It's possible to modify the security groups for an EC2 instance through the *Actions* menu on the instance details page.



To change the subnet and public IPv4 address it's easiest to recreate the instance.

Step 5: Explore more SSH functionality

We barely scratched the surface of SSH functionality. In particular, you will want to learn how to transfer files to your instance using SSH. See **Using ssh with EC2 for login and transferring files (https://canvas.qut.edu.au/courses/20367/pages/using-ssh-with-ec2-for-login-and-transferring-files)** for more information.

Step 6: Terminate the instance

If you are continuing to the next set of exercises for this practical then you can skip this step for now.

Even stopped instances continue to incur AWS charges, so when you complete the exercise, terminate the instance.

1. Use the Terminate option on the Instance state menu to delete the VM.

VSCode ssh remote

SSH-remote in VSCode is handy, but not recommended as it consumes a lot of memory on the VM, which leads to instability. Please use the `ssh` command or Putty instead.