Getting started with the QUT AWS Cloud

In this worksheet you will log in to Amazon Web Services using a QUT student account, create a Linux virtual machine in the cloud, use a secure shell client to connect to the virtual machine, and implement a "Hello World" web application. Finally - and *importantly* - you will terminate the virtual machine.

Note that there are several ways you can connect to the VM, and it is not necessary for you to use all of them. The web browser-based session manager (covered below) is a sensible starting point.

This worksheet links to and relies on instructions on the use of QUT's managed cloud environment hosted on the QUT Wiki. The formatting of these instructions can be confusing at first. Scroll down until you see the how-to description and then it should be straightforward and organised step by step. Please ask us if you have any concerns.

Step 1: Log in to the QUT AWS cloud

After completing this step, you will be ready to access the AWS Management console to create, connect to, and destroy virtual machines.

Background: A dedicated AWS workspace has been created to allow you to access cloud resources using your QUT credentials. This workspace is to be used strictly for academic purposes. The process is described below.

Step by Step Instructions:

- https://wiki.qut.edu.au/display/CDFD/Logging+on+to+QUT+AWS+cloud

Notes:

- After the first time, use this link to access AWS directly: https://d-97671c4bd0.awsapps.com/start#/

Step 2: Create an Ubuntu virtual machine in the QUT AWS cloud

After completing this step, you will have a machine running, ready for remote connection and management. We will continue to use Ubuntu 18.04 for this year. You may use later versions if you wish and if you have some Linux expertise. However, some aspects may be different, and our worksheets will assume 18.04.

Background: You will use an Amazon t2 micro instance running Ubuntu 18.04 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.

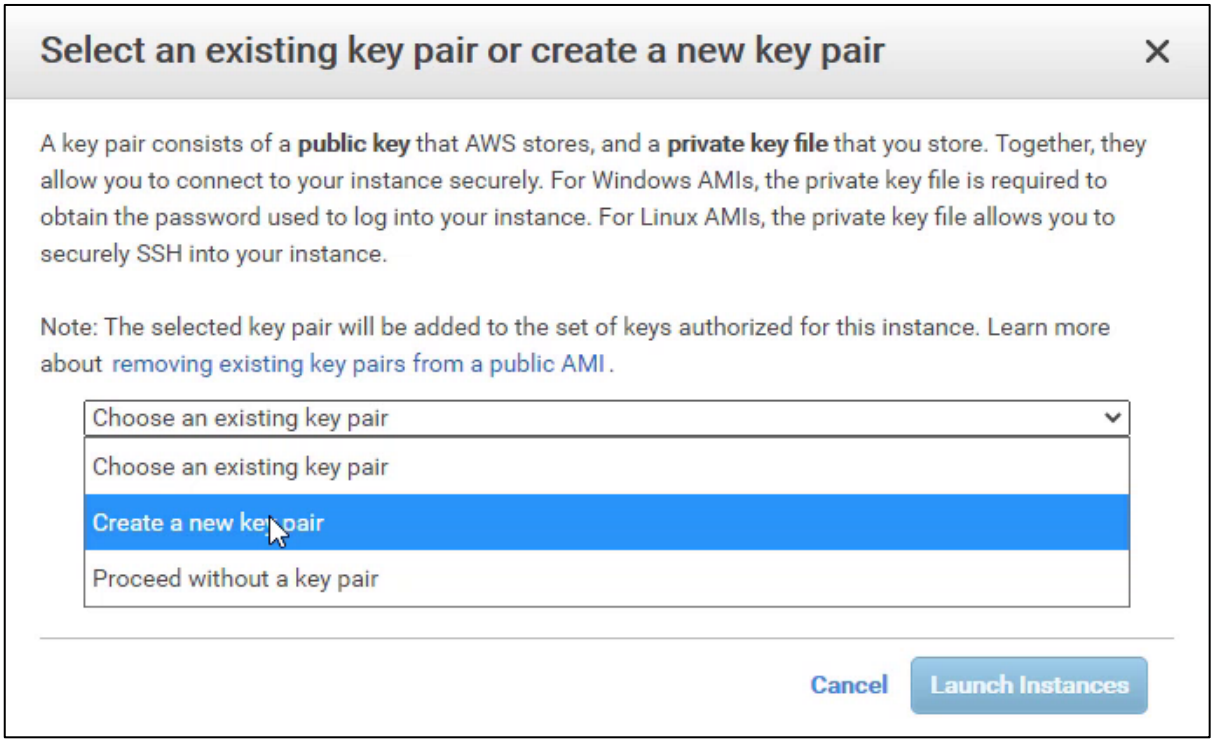
Step by step instructions:

- https://wiki.qut.edu.au/display/CDFD/Building+a+VM+in+AWS

Notes:

- If this is the first time you have created an AWS VM, create and download a private access key (which you save as a .pem file). The key can be used in the next step to establish a remote connection to the VM.

  The system will offer you a choice of existing keys: ignore this and create your own.

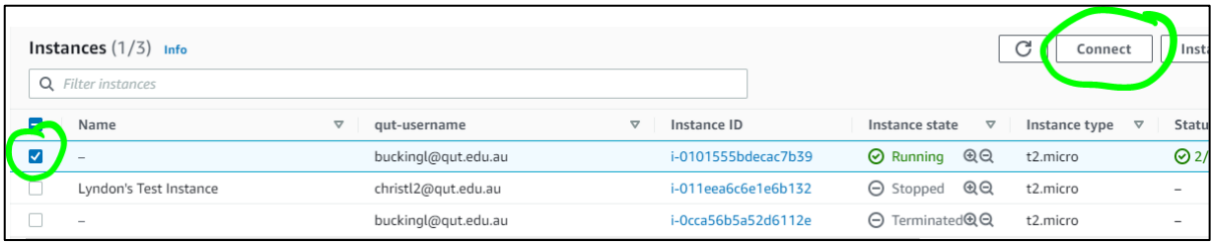


Step 3: Connect to your VM with the web-based Session Manager

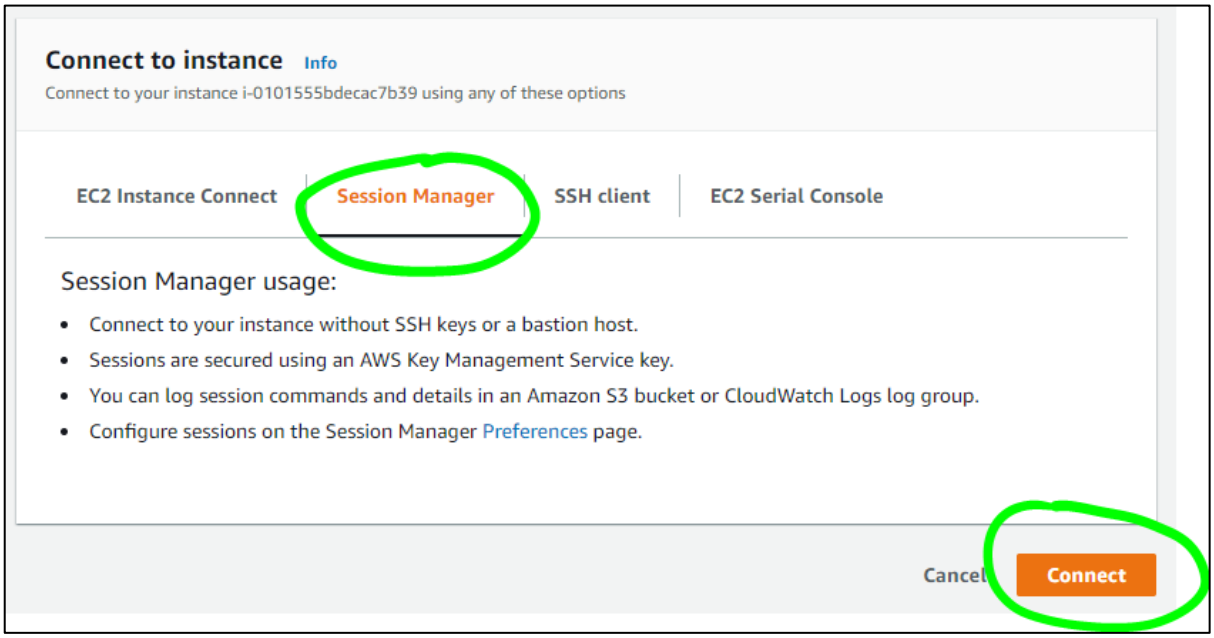
After completing this step, you can work on your VM in a browser tab.

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

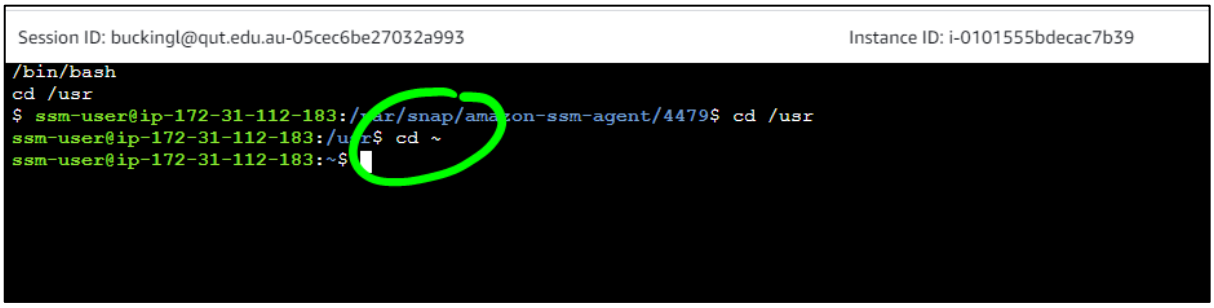
- Start your virtual machine and navigate to the AWS Instances page.
- Select your running instance and click “Connect”.



- Go to the “Session manager” tab, and click “Connect”



- Successful log in will lead to a Linux bash session on the VM, hosted in your browser. Use the command cd ~ to go to your home directory to start work.



At this point you may proceed to Step 6 (hosting a simple HTTP server) should you wish. The intermediary steps explore alternative ways of connecting to the VM.

Step 4: 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. Once you have completed these instructions, Step 5 tells you how to put this into practice to connect to the machine.

Steps 4 and 5 provide an alternative to the web browser session managed approach described in Step 3.

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:

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 .pem file without further processing.

  - The MINGW terminal provides bash, 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 .pem file to a .ppk file for use with Putty.

- Software for MINGW and Putty: Tools.zip

- Installation instructions for all options: BasicTools.pdf

Step 5: Use a secure shell (SSH) client to connect to the running VM

After completing this step, you will be connected to the VM and able to execute commands in the Linux shell.

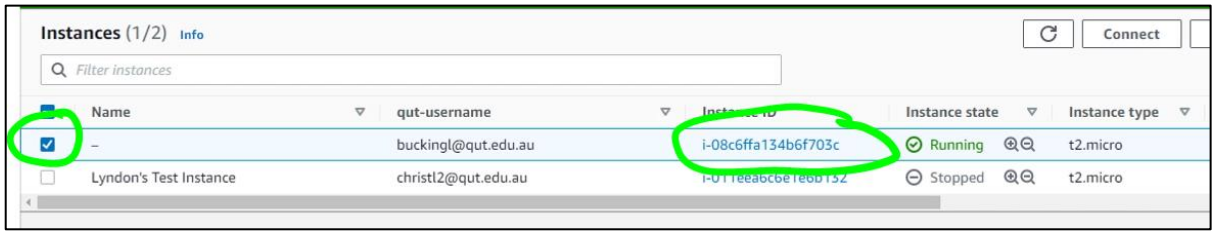
Background: Using the VM's private access key with your SSH client allows you to log on to the VM, install software, and execute programs.

Start a VM via the AWS management console, and make sure you have access to the .pem private key file.

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

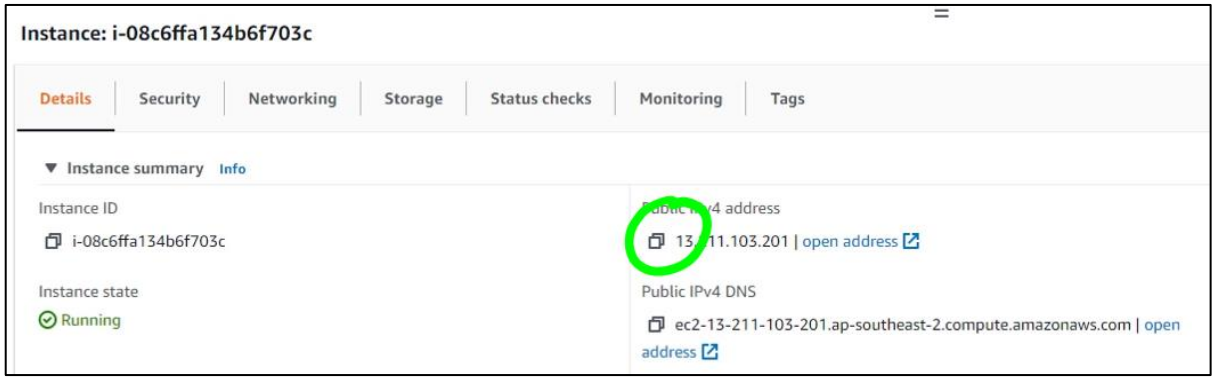
Go to the AWS ECS dashboard, open the list of instances, and select your running instance.



Copy the public IP address of your instance to the clipboard.

In this walk through, the public IP address is

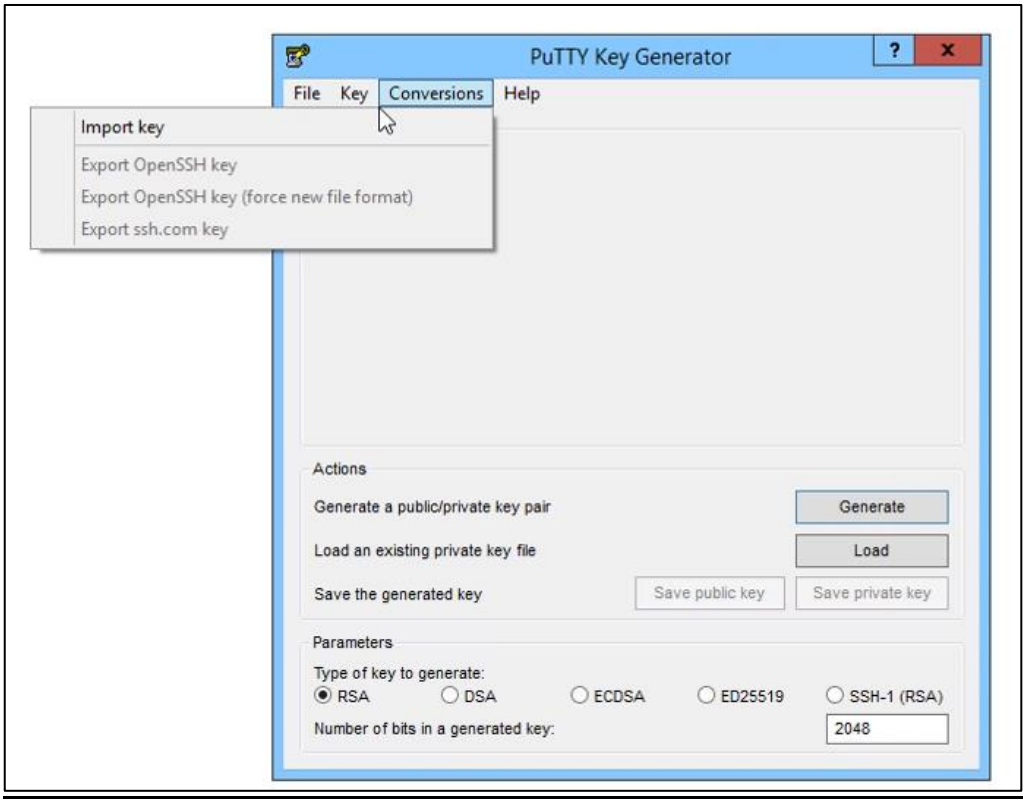
- 13.211.103.201



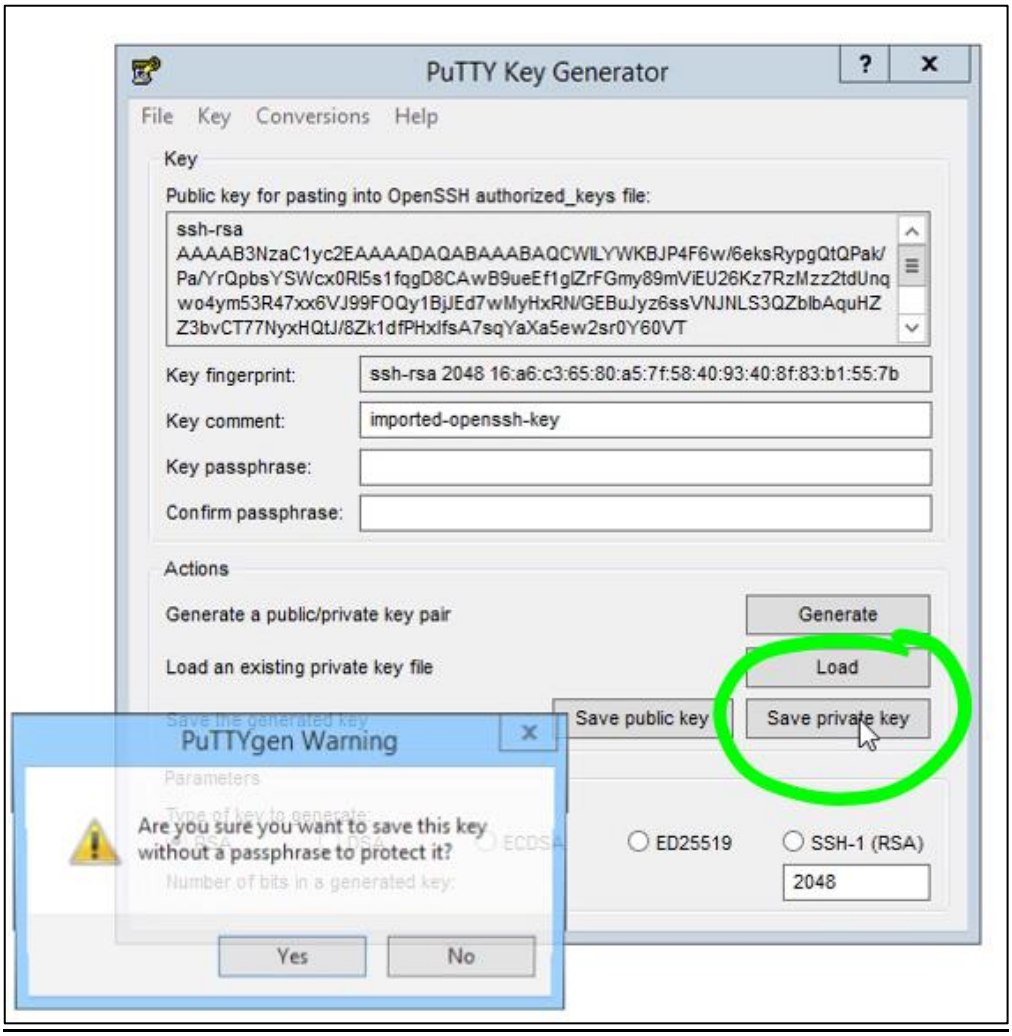
You can connect to the VM using Putty or Bash. Both are now covered in turn.

a) Connecting with Putty

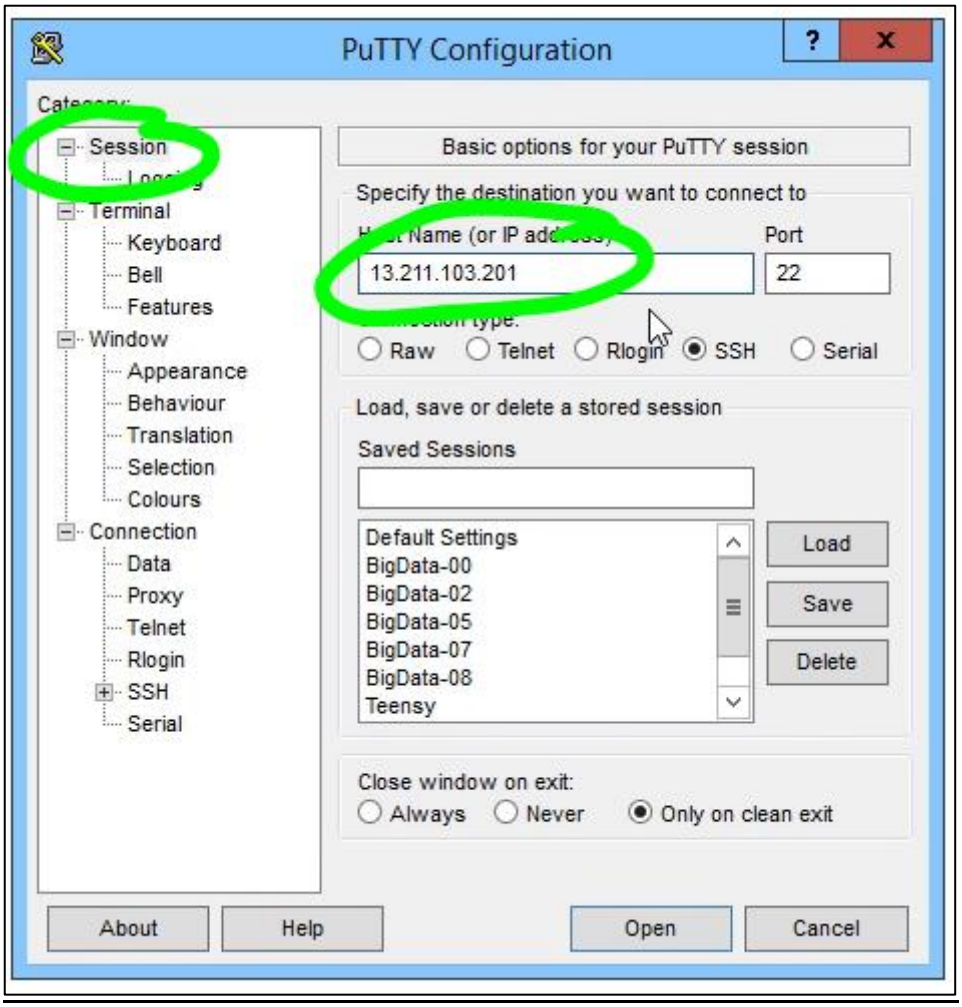
- Run puttygen to convert the OpenSSH key file to a Putty key file
- Import your .pem private key file



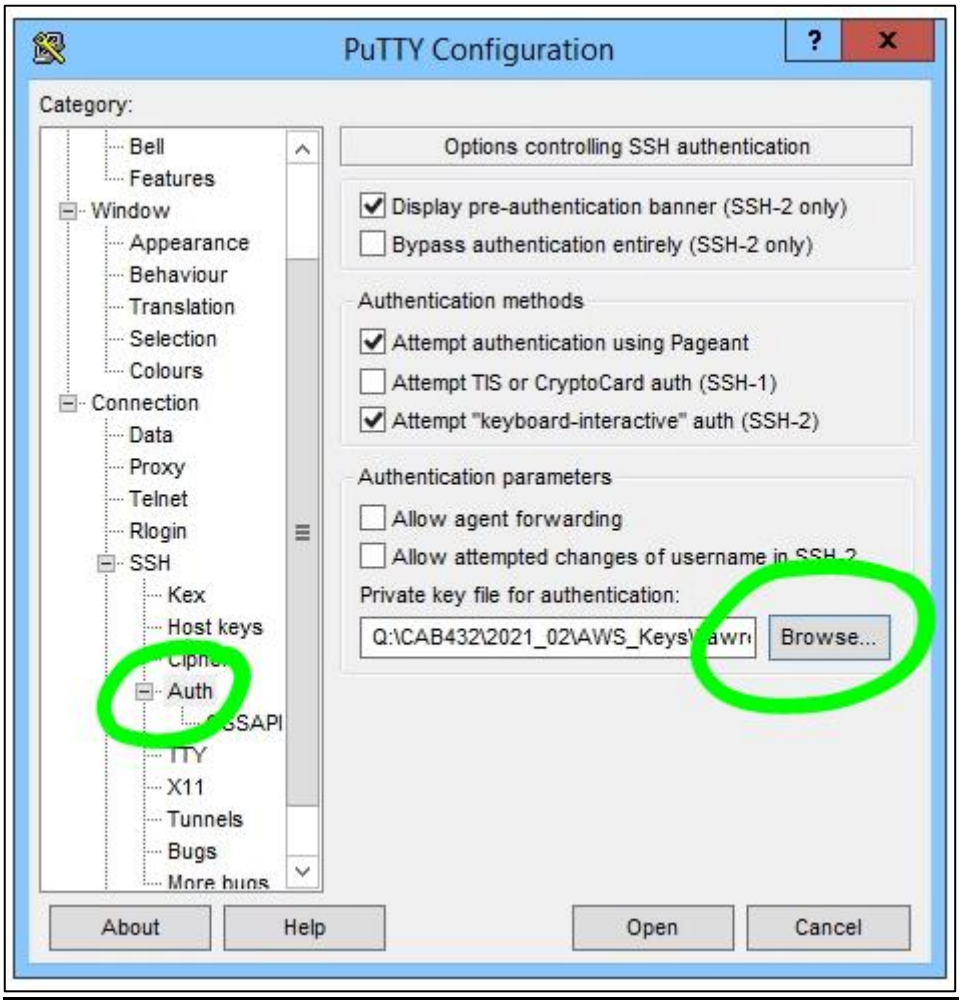
- Save the private key file
  as Q:\CAB432\2021_02\AWS_Keys\LawrenceCAB432_key2.ppk



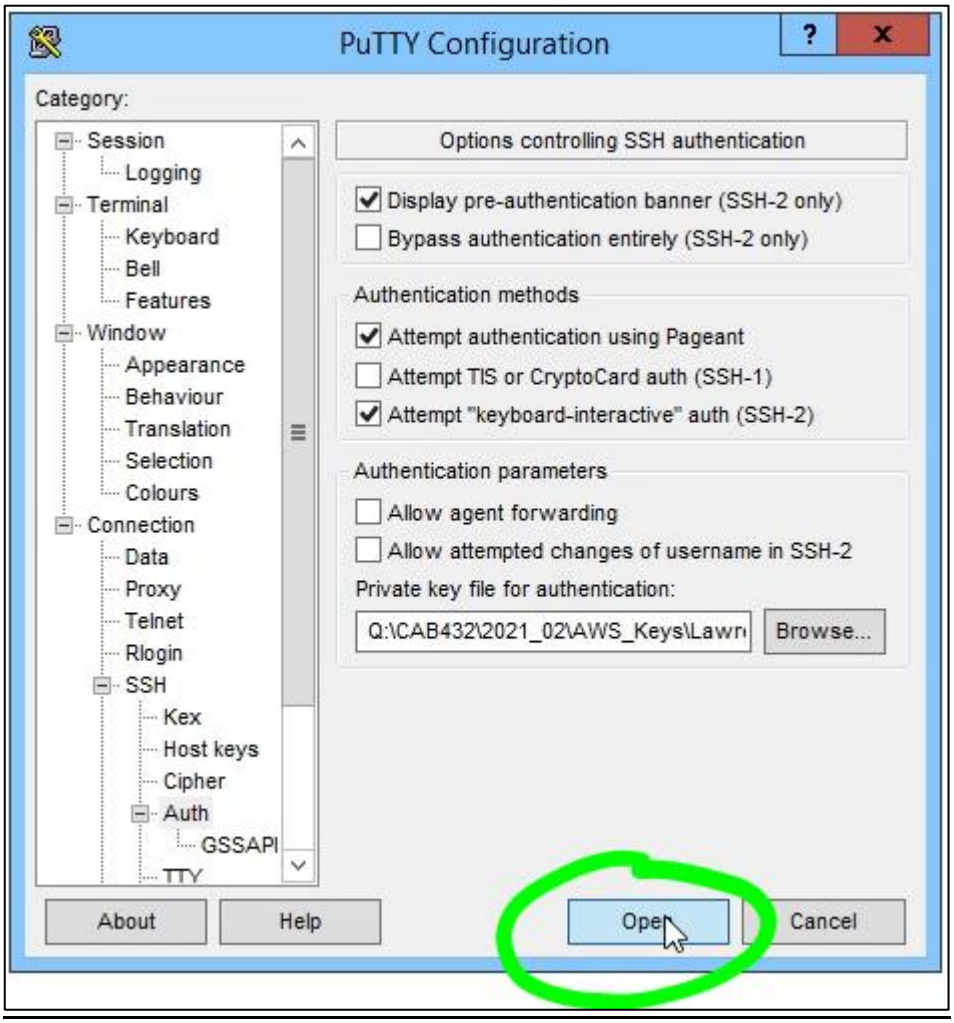
- Run putty.
- On the Session tab, insert the public IP address of the VM.



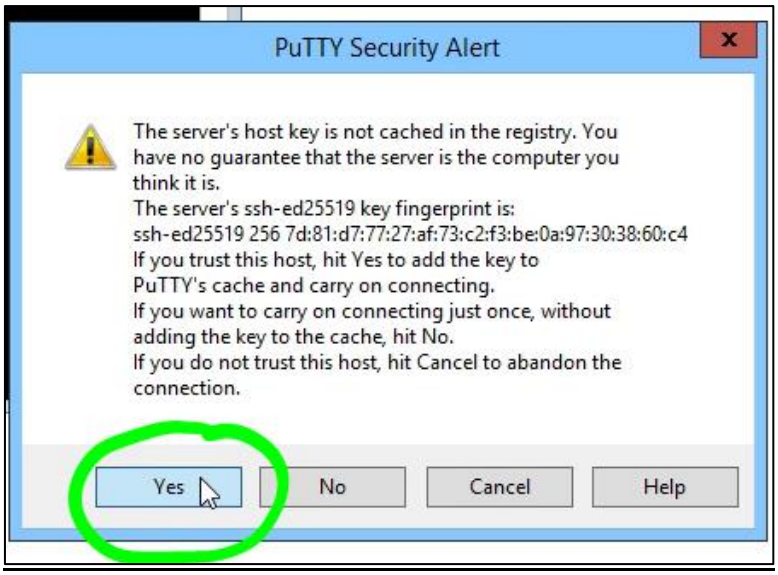
- On the Connection tab, open the private key (.ppk) file.


PuTTY Configuration
Category:
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Kex
Host keys
Auth
TTY
X11
Tunnels
Bugs
Options controlling SSH authentication
Display pre-authentication banner (SSH-2 only)
Bypass authentication entirely (SSH-2 only)
Authentication methods
Attempt authentication using Pageant
Attempt TIS or CryptoCard auth (SSH-1)
Attempt "keyboard-interactive" auth (SSH-2)
Authentication parameters
Allow agent forwarding
Private key file for authentication:
Browse...
About
Help
Open
Cancel

- Click "Open".

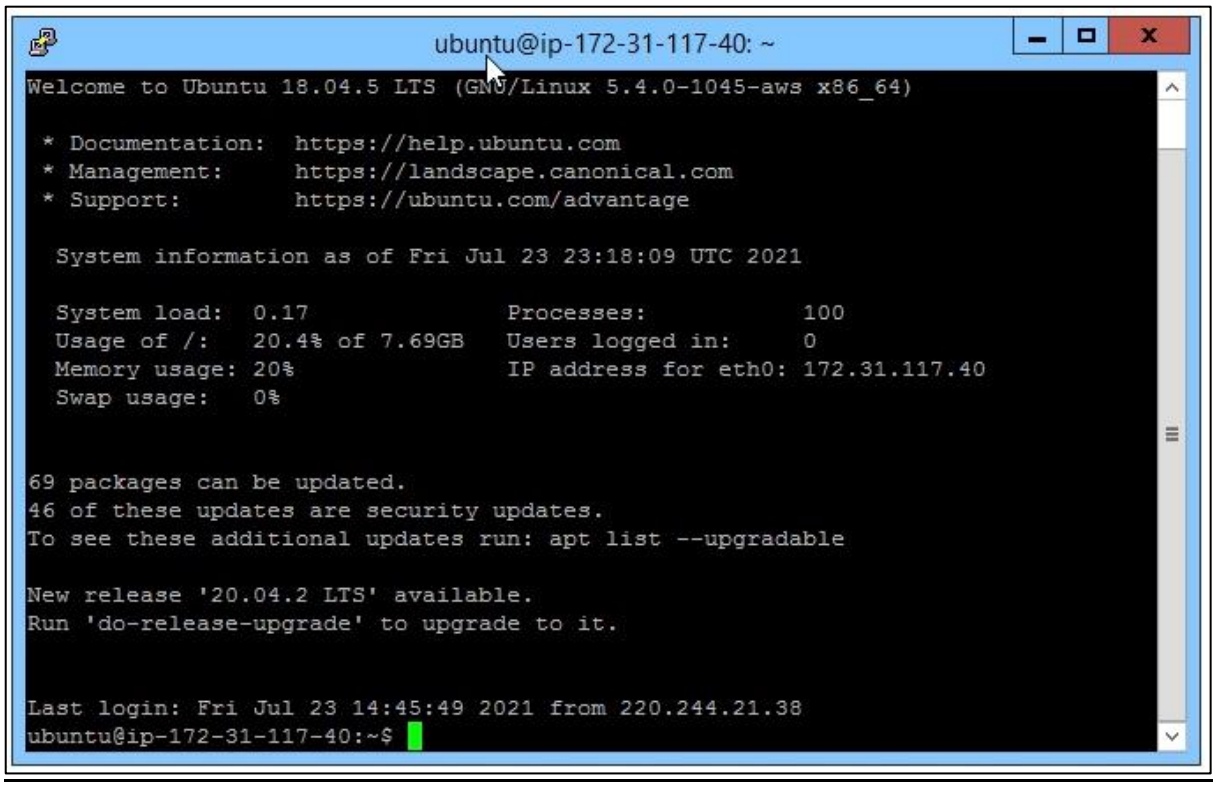


- When prompted to cache the host key, choose "Yes" or "No" (either should be fine).



- When prompted, login as user ubuntu.

- Successful log in will lead to a Linux bash session on the VM.



b) Connecting with Bash

Make sure you have a suitable command line environment set up, e.g., Linux, MacOS, or something like MINGW.

- Open a terminal window with access to ssh
- Use the saved private key file and IP address to connect to the VM. For example:

  ssh -i /Q/CAB432/2021_02/AWS_Keys/LawrenceCAB432_key2.pem -l ubuntu 13.211.103.201

- Replace the file name and IP address with values corresponding to your situation
- When prompted to save the key, enter yes
- Successful log in will lead to a Linux bash session on the VM

Step 6: Host a simple web application on the VM.

After completing this step, you will be able to access a basic web site hosted on the VM.

To undertake this step, you must have connected to the VM using one of the methods described above.

Background: Installing Python lets you host a simple HTTP web site.

- Install Python:

  ```
  sudo apt-get install python-minimal
  ```

- Make sure you're in your home directory:

  ```
  cd ~
  ```

- Create a small HTML document. If you know a Linux editor already you can create the file and include the html given below. If not, this shell echo statement will create a file for you:

  ```
  echo >hello.html '<html><body><h1>Hello CAB432!!!</h1></body></html>'
  ```

- Host the web site:

  ```
  python -m SimpleHTTPServer 3000
  ```

- Stay logged on to the VM and open a new browser tab. Navigate to your document on the VM. In the concrete example used for this walk-through, the address is

  http://13.211.103.201:3000/hello.html

  Replace 13.211.103.201 with the IP address of your own VM.

Step 7: Terminate the instance

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

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