CS 260
How to install & setup Amazon EC2

You can use the following steps to create a web server that you can use for the labs for the rest of the semester. You will be turning in a URL that will access your application on your EC2 server.

During this process you will become the administrator of a Linux machine and will have superuser permissions. Along with learning how to build your web application, you ought to spend some time learning system administration concepts. So, if things seem unfamiliar, ask questions in class.

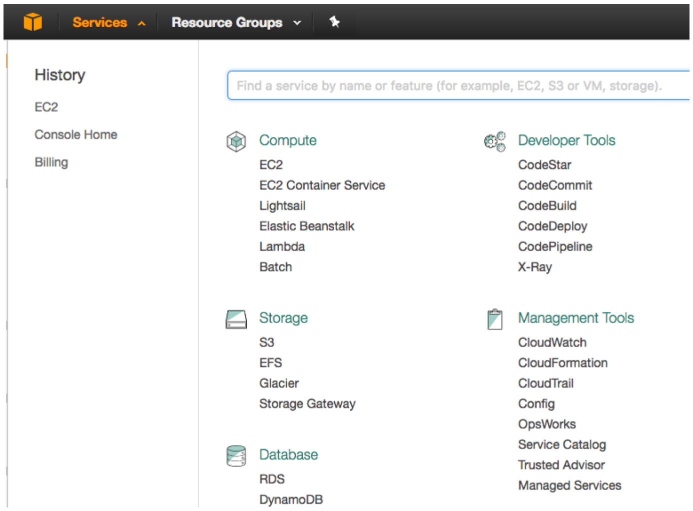
The following steps should get you up and running. Try to follow them closely, then you can come back later and explore.

Create an Amazon Web Services (AWS) account

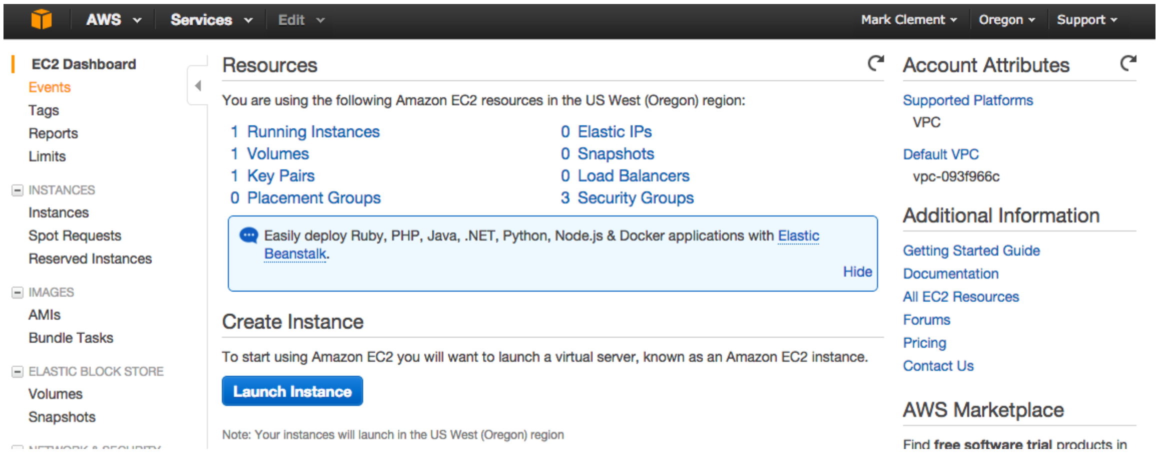
If you don't have one already, create an Amazon Web Services (AWS) account

Launch an EC2 instance (virtual server)

In order to install and run your instance, we first need to have a virtual server. Amazon calls its virtual servers *instances*, because you can have many of them running at the same time. For now, you only need one instance.
Login with your AWS account and goto the EC2 button in the upper left hand corner.



Then select the button to launch an instance



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

1 Running Instances	0 Elastic IPs
1 Volumes	0 Snapshots
1 Key Pairs	0 Load Balancers
0 Placement Groups	3 Security Groups

Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with Elastic Beanstalk.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US West (Oregon) region

Account Attributes

Supported Platforms
VPC

Default VPC
vpc-093f966c

Additional Information

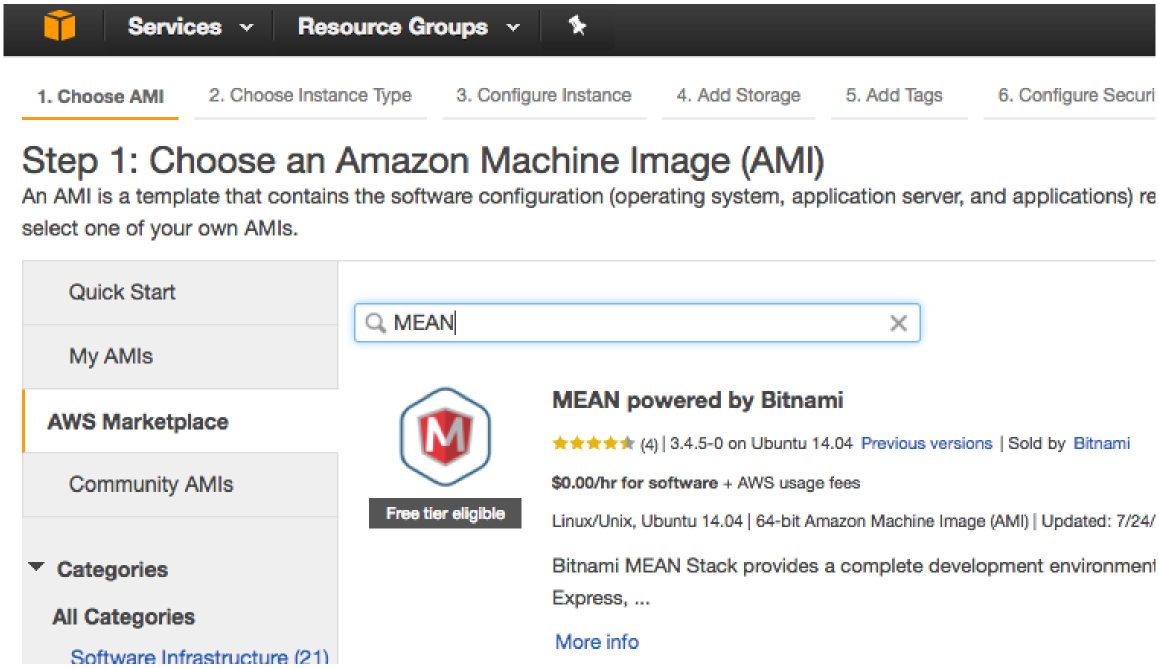
Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

AWS Marketplace

Find **free software trial** products in

Step 1: Choose the Bitnami image

When you launch an instance, you will have the chance to initialize the instance with a set of applications and configurations. Pick the AWS Marketplace on the left menu, then search for MEAN in the search bar. You should see "MEAN powered by Bitnami (HVM)". Select it and it will take you to Step 2.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) re select one of your own AMIs.

MEAN powered by Bitnami

★★★★★ (4) | 3.4.5-0 on Ubuntu 14.04 Previous versions | Sold by Bitnami

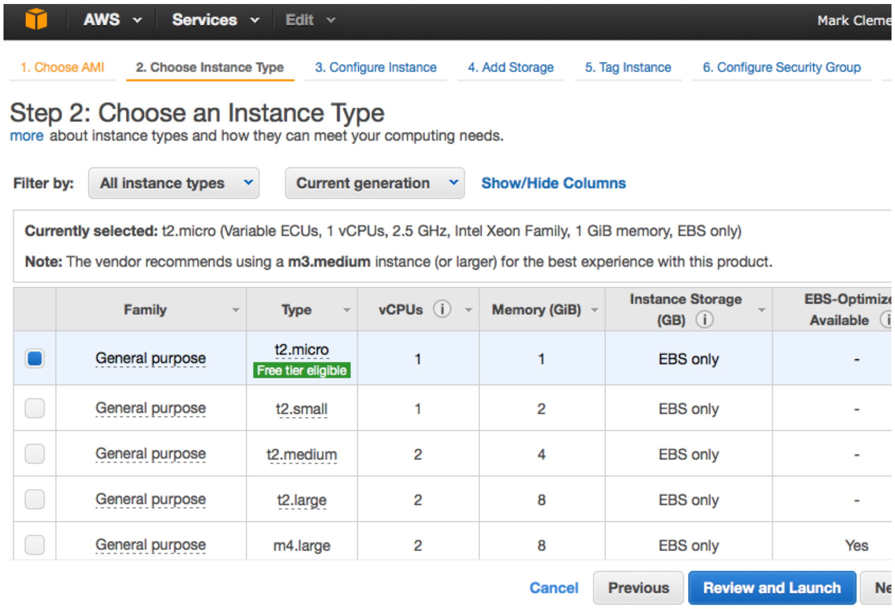
$0.00/hr for software + AWS usage fees

Linux/Unix, Ubuntu 14.04 | 64-bit Amazon Machine Image (AMI) | Updated: 7/24/

Bitnami MEAN Stack provides a complete development environment Express, ...

Step 2: Launch the Instance

Once you select this instance, select the MEAN image, choose the t2.micro option since it is the only free tier option. And select the “Review and Launch” button in the bottom right part of the screen.



Step 2: Choose an Instance Type

more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

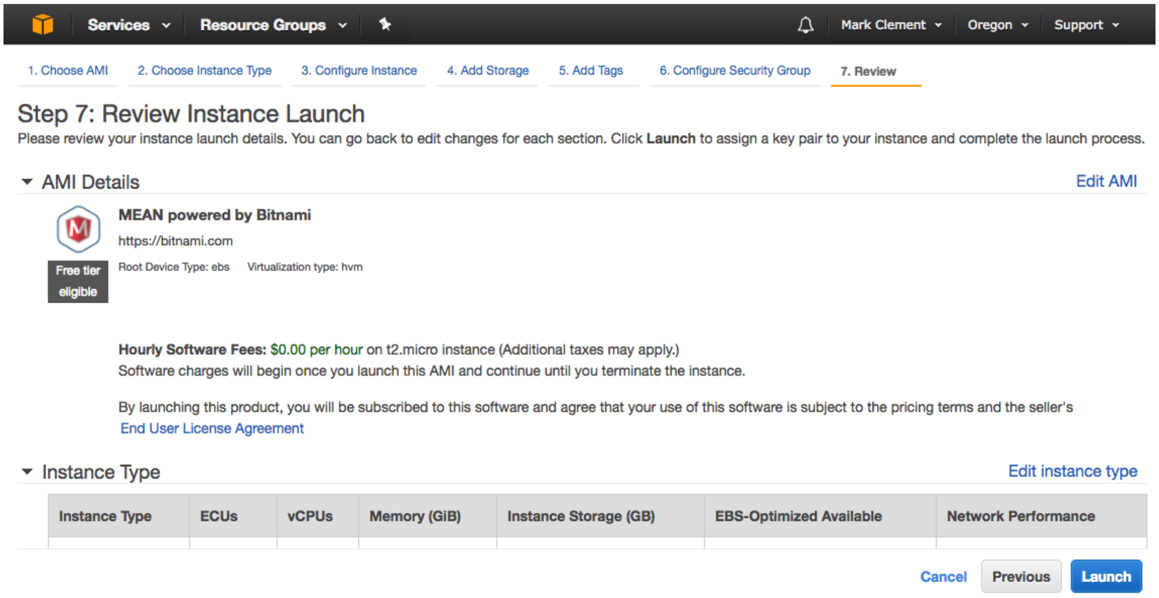
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

Note: The vendor recommends using a **m3.medium** instance (or larger) for the best experience with this product.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimiz Available
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-
□	General purpose	t2.small	1	2	EBS only	-
□	General purpose	t2.medium	2	4	EBS only	-
□	General purpose	t2.large	2	8	EBS only	-
□	General purpose	m4.large	2	8	EBS only	Yes

Step 7: Review Instance Launch

press the “Launch” button



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

MEAN powered by Bitnami

https://bitnami.com

Root Device Type: ebs   Virtualization type: hvm

Hourly Software Fees: $0.00 per hour on t2.micro instance (Additional taxes may apply.)

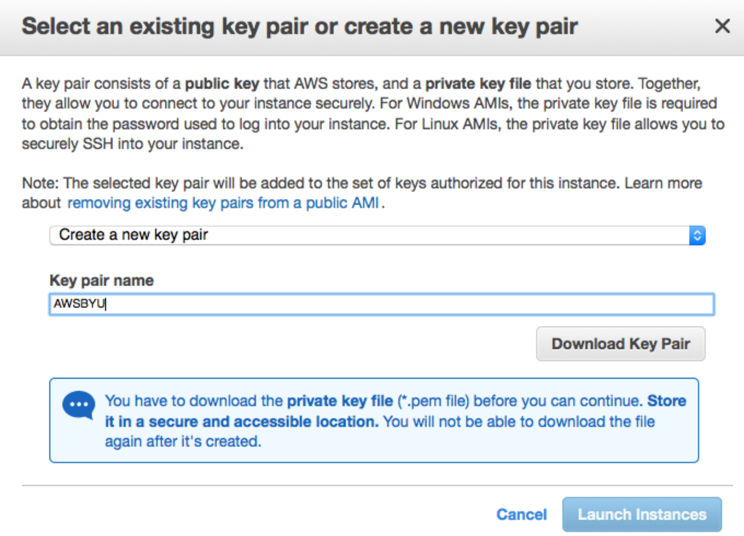
Software charges will begin once you launch this AMI and continue until you terminate the instance.

By launching this product, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's End User License Agreement

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance

Now you will create a security key that will be used to access your virtual machine. Give it a name and make sure it is downloaded to your laptop or virtual machine. Then click “Launch Instance”.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

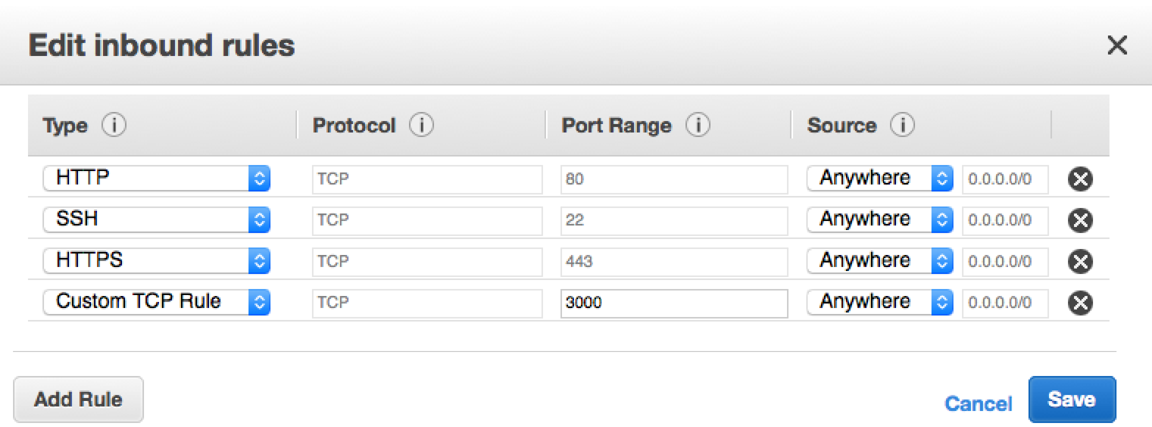
Create a new key pair

Key pair name: AWSBYU

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

It will take a while to configure your machine. You can take a look at the documentation, then go to “View Instances” in the bottom right corner. The status field will say “initializing” for a while until the machine is ready for you to use.

While you are waiting, select the “Network and Security”/”Security Groups” tab on the left menu. Select the group with “MEAN” in the name, and make sure it includes port 3000-3010 for inbound traffic. If it doesn’t, then add it with “Actions”/”Edit Inbound rules”.



Edit inbound rules

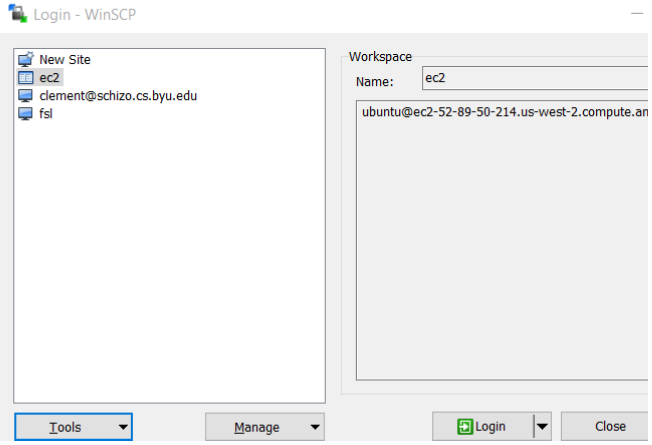
Type	Protocol	Port Range	Source	
HTTP	TCP	80	Anywhere	0.0.0.0/0
SSH	TCP	22	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0
Custom TCP Rule	TCP	3000	Anywhere	0.0.0.0/0

You can now use your instance’s public dns to connect to your server via ssh. Amazon does not provide a custom username and password for ssh connections. Instead, they use the key pair file you created a couple of steps ago. Note the public DNS and public IP addresses that are shown in this screen. You will need them to connect to your server

Connect to Amazon EC2

If you’re on Windows, you can use winscp:

1. Download Putty and puttygen with winscp
2. Start puttygen from the winscp tools menu

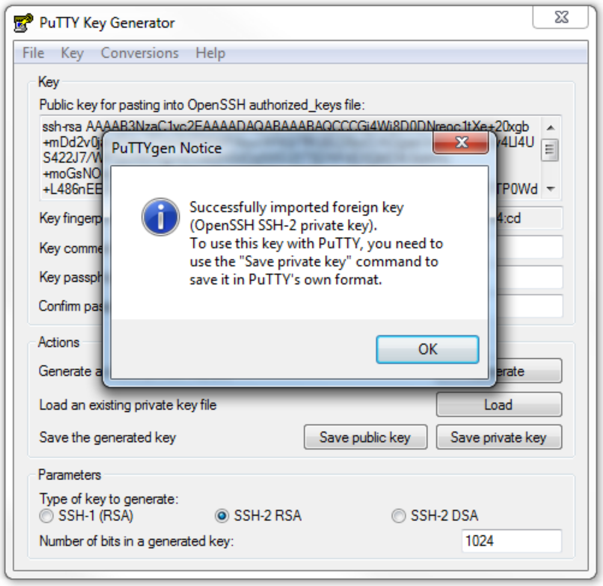


Name: ec2

ubuntu@ec2-52-89-50-214.us-west-2.compute.an

3. Use puttygen to convert Amazon’s .pem key pair file to .ppk file.
   - From puttygen select **Load**

- Select *view all files* and pick the .pem file you download from Amazon.



PuTTY Key Generator

Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCCCGi4Wi8D0DNreoc1tXe+20xgb
+mDd2v0j                                                    4U4U
S422J7/W
+moGsNO
+L486nEE                                                   TP0Wd
```

Key fingerp   4:cd

Key comme

Key passph

Confirm pas

PuTTYgen Notice

Successfully imported foreign key (OpenSSH SSH-2 private key).
To use this key with PuTTY, you need to use the "Save private key" command to save it in PuTTY's own format.

Actions

Generate a   erate

Load an existing private key file

Save the generated key

Parameters

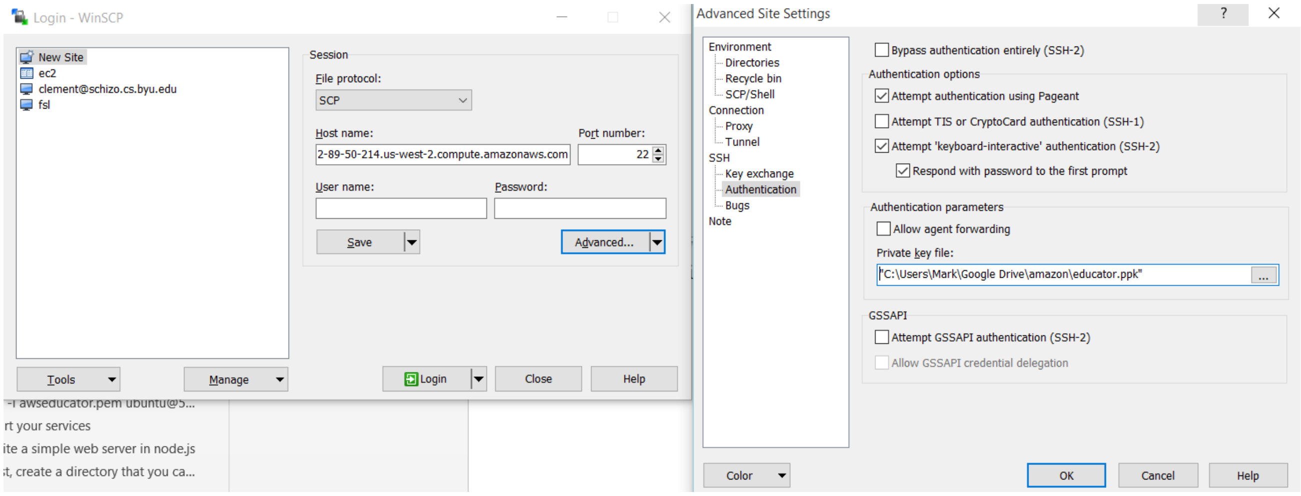
Type of key to generate:

○ SSH-1 (RSA)   ● SSH-2 RSA   ○ SSH-2 DSA

Number of bits in a generated key: 1024

- Click **OK** and select **Save Private Key** A passphrase is not required.

4. Now connect to your instance using winscp
    - Enter public dns address for your instance (for example http://ec2-52-89-50-214.us-west-2.compute.amazonaws.com/)
    - Navigate to Connection/SSH/Auth. Click **Browse** and select the .ppk file you exported from puttygen.
    - Login with “User name” ubuntu



Login - WinSCP

Session

File protocol: SCP

Host name: 2-89-50-214.us-west-2.compute.amazonaws.com

Port number: 22

User name:

Password:

-i awseducator.pem ubuntu@5...

rt your services

ite a simple web server in node.js

t, create a directory that you ca...

Advanced Site Settings

□ Bypass authentication entirely (SSH-2)

Authentication options

☑ Attempt authentication using Pageant

□ Attempt TIS or CryptoCard authentication (SSH-1)

☑ Attempt 'keyboard-interactive' authentication (SSH-2)

☑ Respond with password to the first prompt

Authentication parameters

□ Allow agent forwarding

Private key file: "C:\Users\Mark\Google Drive\amazon\educator.ppk"

GSSAPI

□ Attempt GSSAPI authentication (SSH-2)

□ Allow GSSAPI credential delegation

- Now, you should be able to open a command line terminal through “Commands/Open in Putty”

If you get a message like “Putty doesn’t exist on your computer”, then download and install putty
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

If you are running Windows 10 you can use Linux for Windows

Enabling it is simple:
>Open Settings
>Click on Update and Security
>Click on For Developers on left side bar
>Select Developer Mode to turn on developer mode
>Restart computer
>Open Control Panel
>Click on Programs
>Click on Turn Windows features on or off
>Scroll down to Windows Subsystem for Linux and click ok
>Restart computer

>Open Start and search for bash.exe
>On command prompt type y and press Enter to download and install bash
>It will ask you to create a UNIX user account. It does not have to be the same as your Windows account

After that you can follow the same instructions to ssh as you would for Linux. You can also use it in any other way you would use Linux (use command-line tools and applications, run bash scripts, modify files add languages etc).

I highly suggest looking into it for the student's who use Windows because it is just like having Linux on your computer. I no longer have to have a second computer just for Linux.

https://msdn.microsoft.com/en-us/commandline/wsl/about

Apple or Linux

You will first need to change the permissions on your .pem file.

```
chmod 600 ~/Downloads/clement.pem
```

If you are using an apple or linux machine, you will use ssh instead of putty.

Open a terminal window and run ssh with a command like the following, but replace the 52.24.3.226 with the IP address or URL for your instance.

```
ssh -i awseducator.pem ubuntu@52.24.3.226
```