In this guide, we will be using AWS Free tier to setup and SSH into a Linux Ubuntu server. Keep in mind that AWS Free tier only lasts for 1 year. After this, you will start getting charged for any running instances.

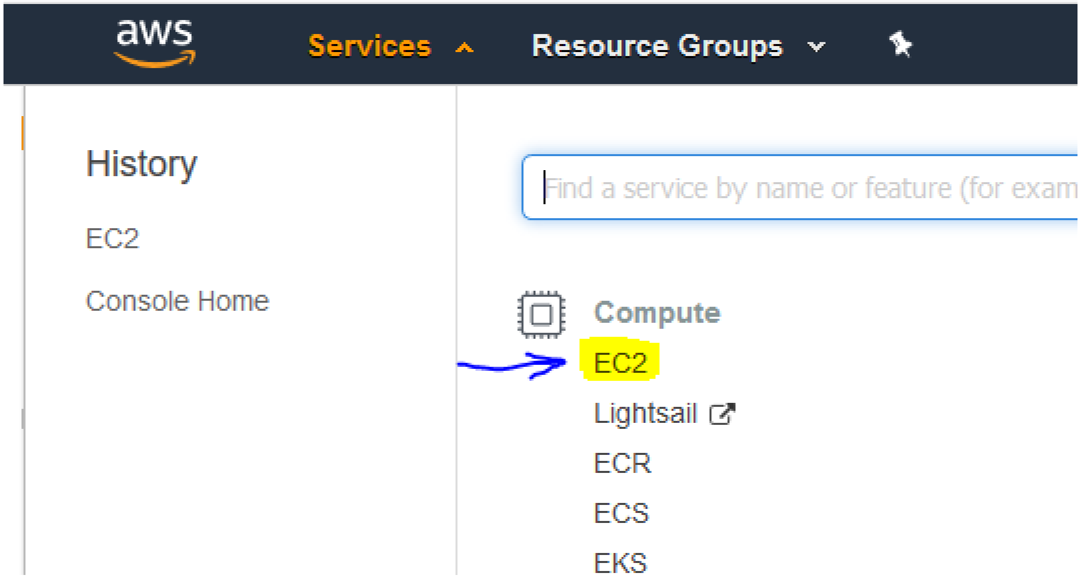
Prerequisites:

- AWS account with attached payment information
- PuTTY and PuTTYgen. Download can be found at https://www.PuTTY.org (you will be navigated to a different site to download it, but it is safe)

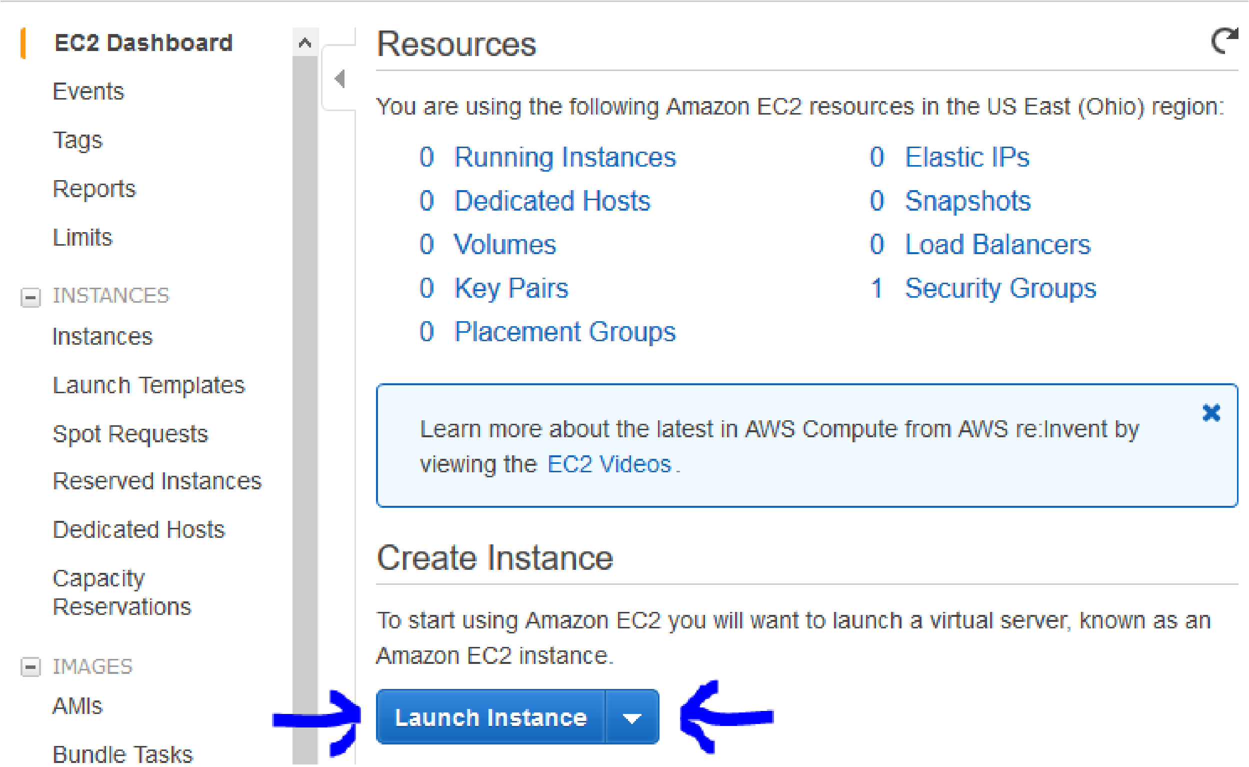
Time needed: 1 hour

1) Create AWS account

- Navigate to https://aws.amazon.com
- Click on My Account at the top and Choose 'AWS Management Console'
- Hover over Services and select EC2



2) Click on ‘Launch Instance’



Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

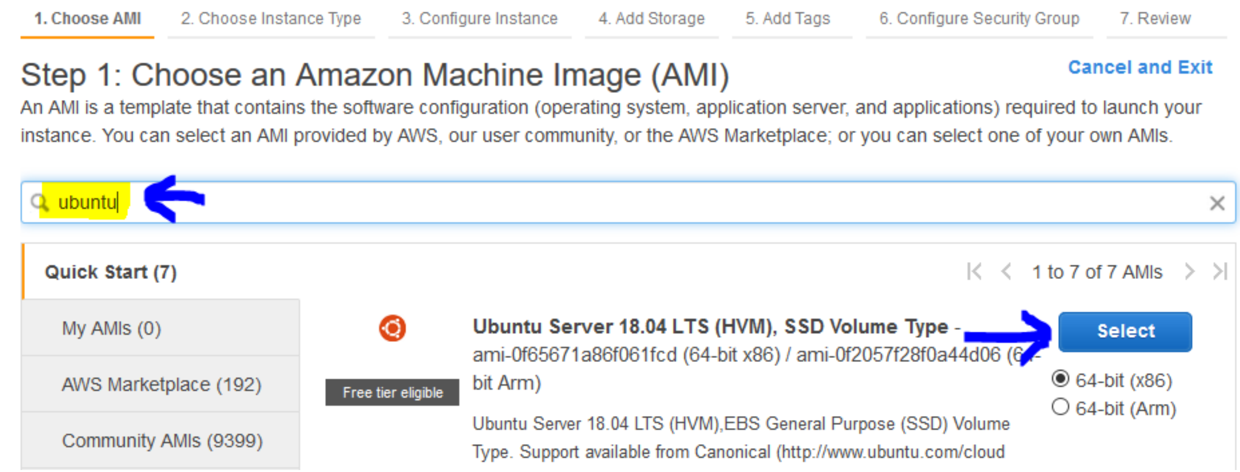
0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
0 Key Pairs	1 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.

Create Instance

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

3) Type ‘ubuntu’ in the search bar and then select the Ubuntu Server 18.04 (Free tier eligible)



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

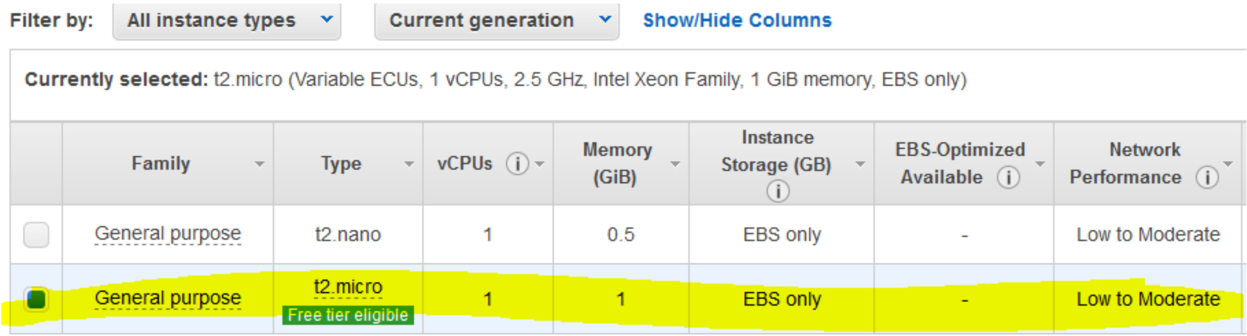
ubuntu

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0f65671a86f061fcd (64-bit x86) / ami-0f2057f28f0a44d06 (64-bit Arm)

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud

● 64-bit (x86)
○ 64-bit (Arm)

4) The Free tier option should be chosen by default.
Click Next: Configure Instance Details



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

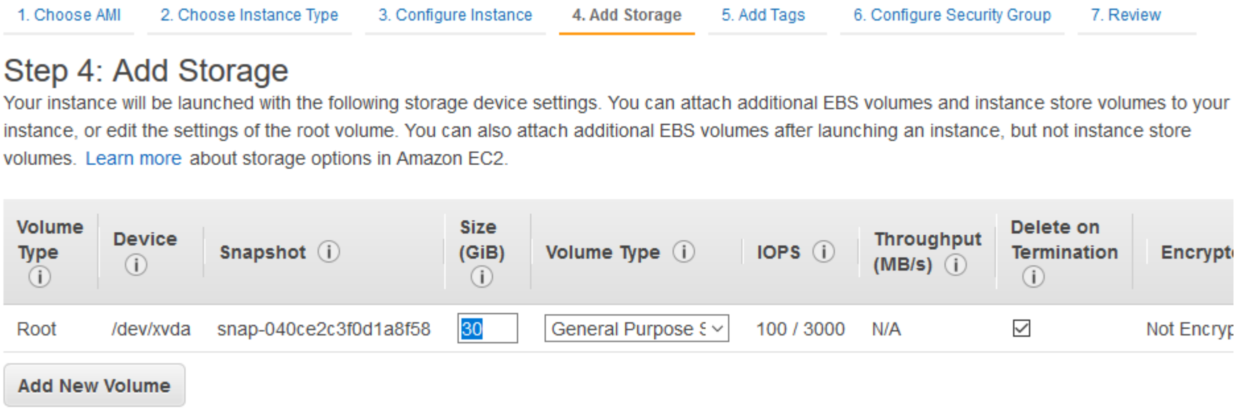
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate

5) Configure Instance: No need to change anything here. Click Next: Add Storage

6)

- You currently get 30GB of storage on free tier. Since you can only have one EC2 instance running 24/7 for free, I choose to change this from 8GB to 30GB.
- Click Next: Add Tags



1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

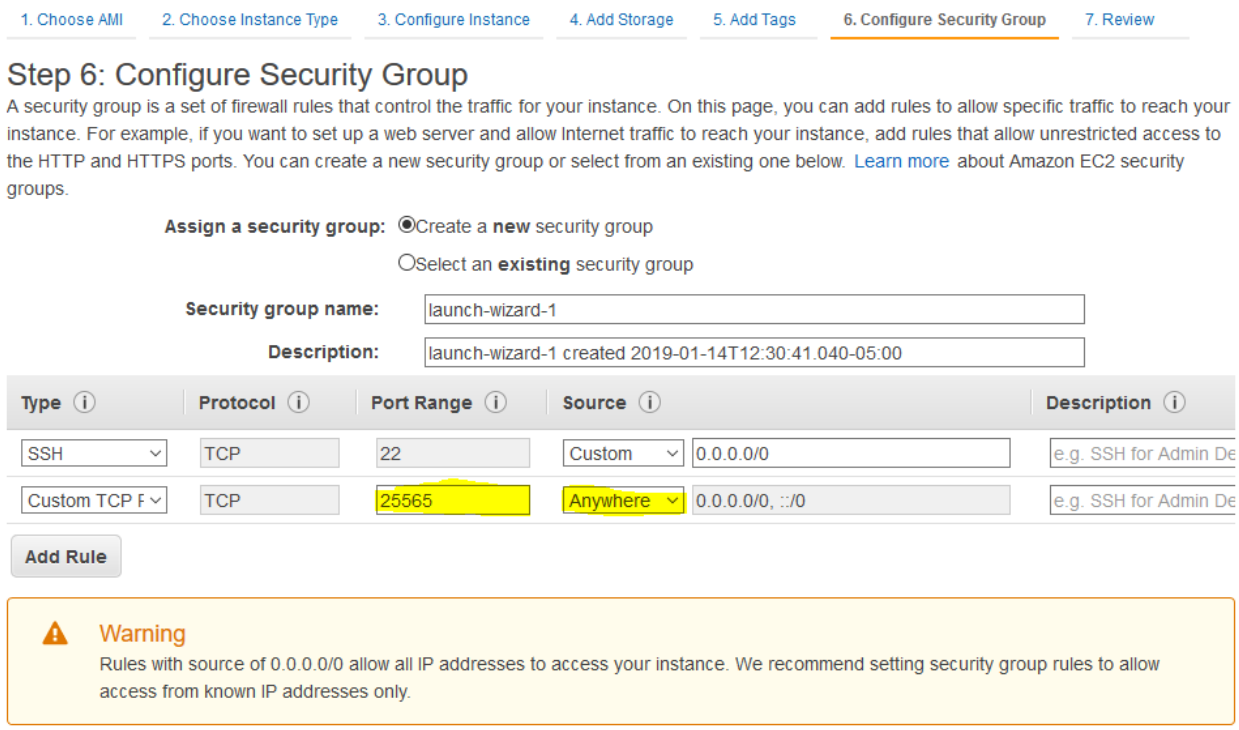
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypt
Root	/dev/xvda	snap-040ce2c3f0d1a8f58	30	General Purpose S	100 / 3000	N/A	☑	Not Encryp

Add New Volume

7) Click Next: Configure Security Group. Click ‘Add Rule’ button and input 25565 in ‘Port Range’ and change ‘Source’ to ‘Anywhere’



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group

○ Select an **existing** security group

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2019-01-14T12:30:41.040-05:00

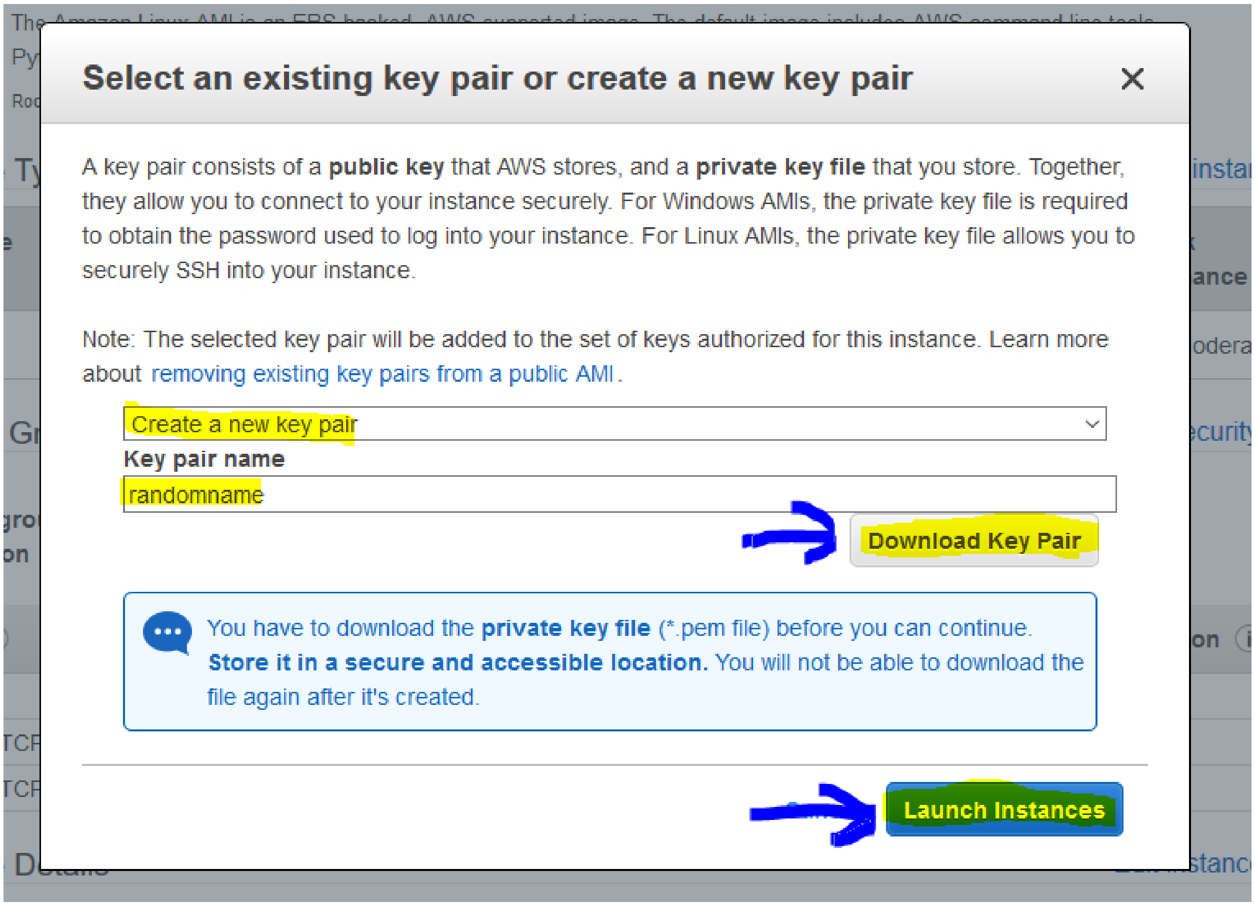
Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin De
Custom TCP F	TCP	25565	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin De

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

8) Click Review and Launch

- Verify everything is correct
- Click ‘Launch’. You will get the screen below
- Select ‘Create a new key pair’ and name it whatever you want
- Click Download Key Pair and save it somewhere you can access it
- Click Launch Instances
- Click View Instances



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

randomname

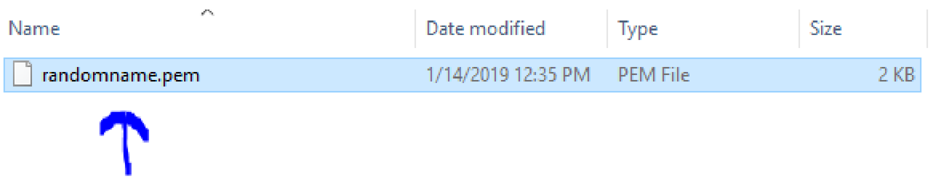
You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

SSH To Instance:

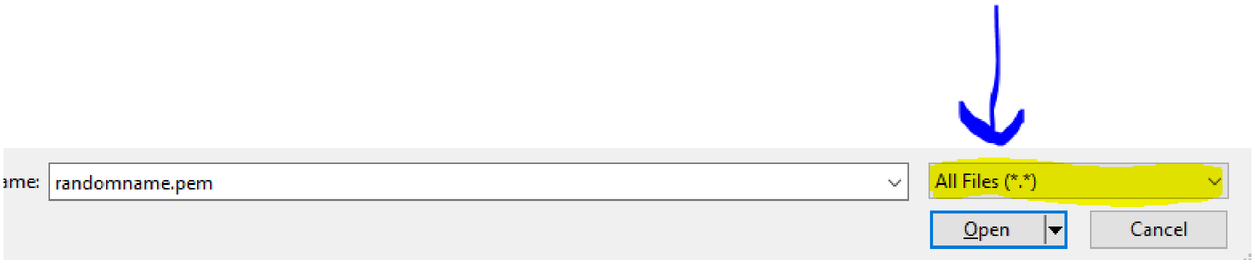
**The following steps 9 & 10 are for Windows users only. If you aren't using Windows or prefer a different guide, check out Amazon's guides here: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html

9) Open PuTTYgen.exe

- Click 'Load' and navigate to the folder where you downloaded the Key Pair to
- If using Windows, make sure select 'All Files' in the bottom right in order to see your key
- Choose your key and click 'Open'. You should get message "Successfully imported foreign key…"



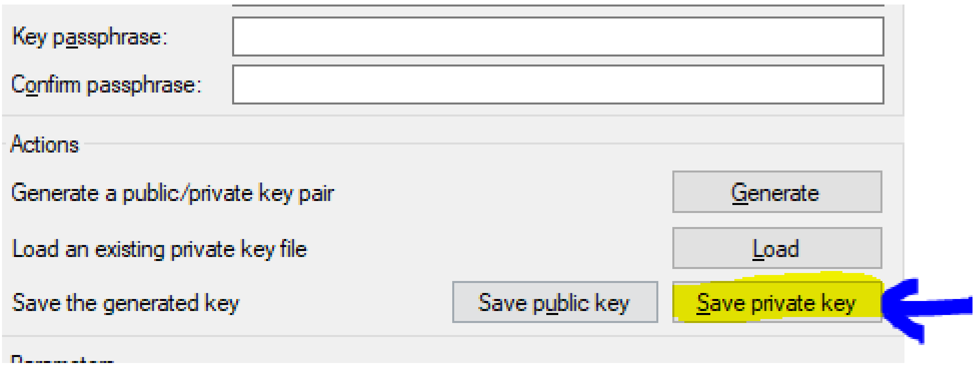
Name	Date modified	Type	Size
randomname.pem	1/14/2019 12:35 PM	PEM File	2 KB



ame: randomname.pem

All Files (*.*)

- Now click 'Save private key'. You can add a passphrase if you want, but I usually don't. If you are worried about someone getting their hands on your key, then you should add a passphrase.



Key passphrase:

Confirm passphrase:

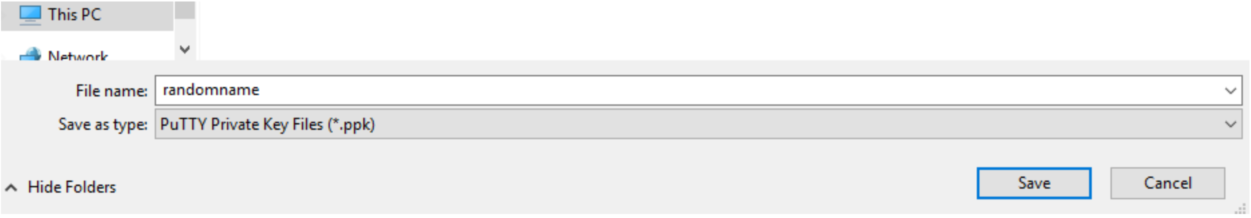
Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

- Name this new key. I usually use the same name as the .pem file. This file should save as a .ppk
- Click Save

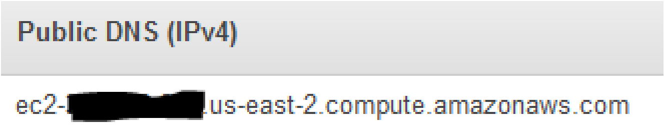


File name: randomname

Save as type: PuTTY Private Key Files (*.ppk)

10) Open PuTTY.exe

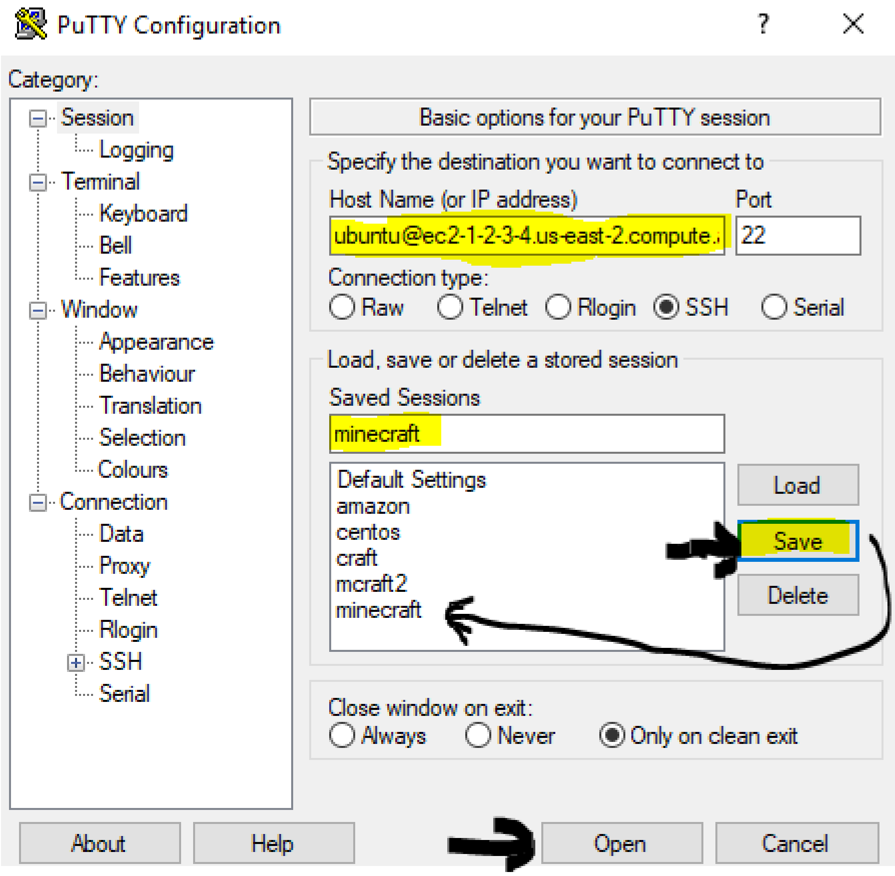
- Expand ‘SSH’ towards the bottom. Then click on ‘Auth’
- On the right hand side, click ‘Browse’, and select your newly created .ppk key file
- Scroll back up to the top of PuTTY and click on ‘Session’.
- Go back to your browser where we created the Ubuntu server on aws, and locate the instance we just created (Services >> EC2 >> Instances (or ‘Running instances’)) . You should see your instance here, and what we are looking for is the Public DNS. Copy this field, starting with ‘ec2’ and ending with ‘.com’ which should look similar to the below picture



Public DNS (IPv4)

ec2-[redacted].us-east-2.compute.amazonaws.com

- Tab back to PuTTY. In the ‘Host Name (or IP address)’ box , paste the DNS. Then, prepend ‘ubuntu@’ to the DNS. So, if your DNS is ec2-1-2-3-4.us-east-2.compute.amazonaws.com, your box should look like ubuntu@ec2-1-2-3-4.us-east-2.compute.amazonaws.com
- You can save this config for future use by giving it a name under ‘Saved Sessions” and clicking Save (shown below)
- Finally, click Open



Specify the destination you want to connect to

Host Name (or IP address): ubuntu@ec2-1-2-3-4.us-east-2.compute.

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

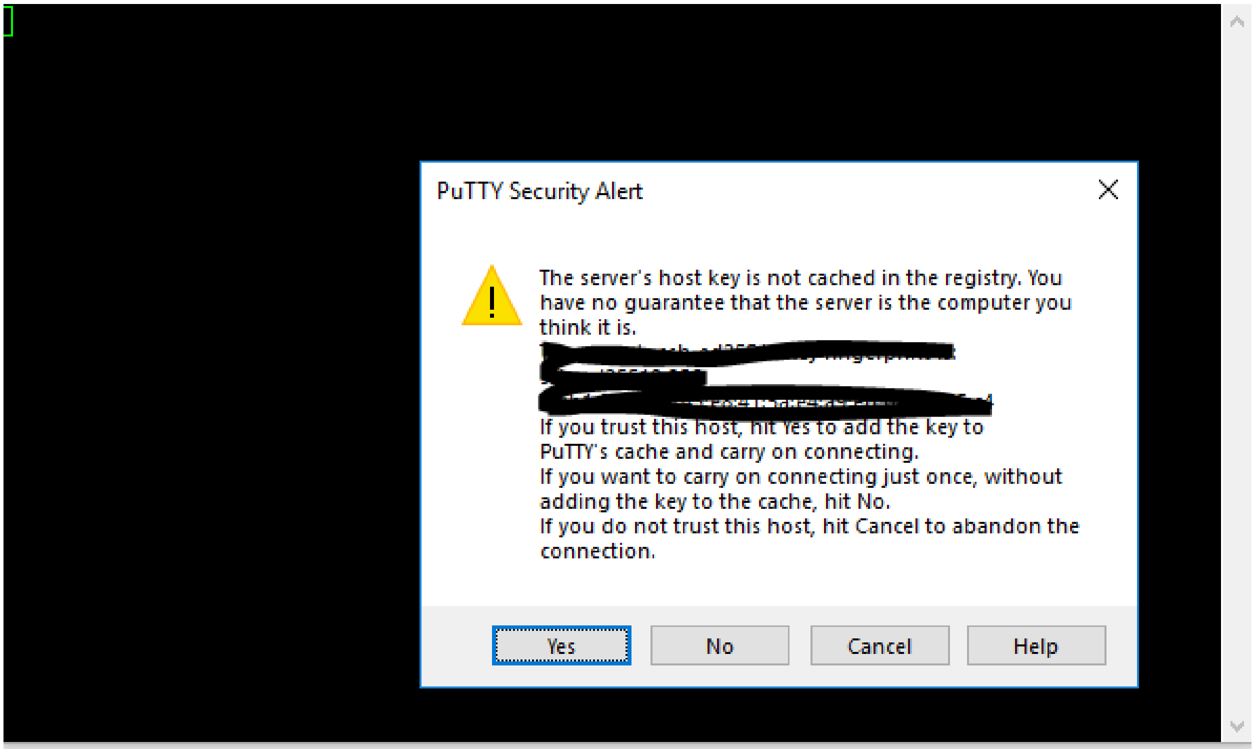
Load, save or delete a stored session

Saved Sessions: minecraft

Default Settings
amazon
centos
craft
mcraft2
minecraft

Close window on exit: ○ Always ○ Never ● Only on clean exit

- You will receive the below security alert. Click ‘Yes’



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
[redacted]
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

- ➢ For more info on connecting to your server instance from Windows using PuTTY, check out Amazon's guide here https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html

By now you should be connected to your server. If you had any issues following this guide, try looking at Amazon's User Guide here: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html