

# 15-441/641: Recitation 1

TAs: Nirav Atre, Zili Meng, Hugo Sadok

09/02/2022

## 1 Setting up an AWS Instance

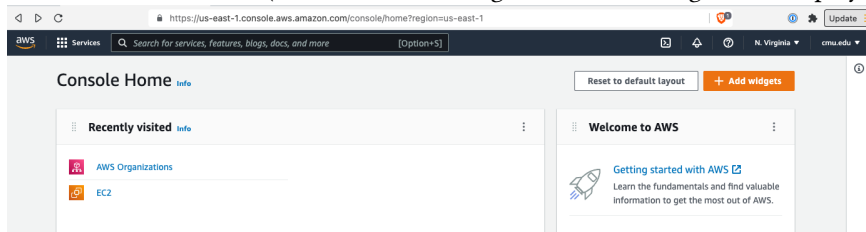
To start, you need an Amazon Web Services (AWS) Account. You should have received an email yesterday with an invitation to create an AWS account.

**Warning:** A lot of stuff you do on AWS will be charged. Be mindful of launching 100 instances, it's cool but expensive. You are also charged depending on the duration, so leaving a machine running indefinitely will also cost you more credits.

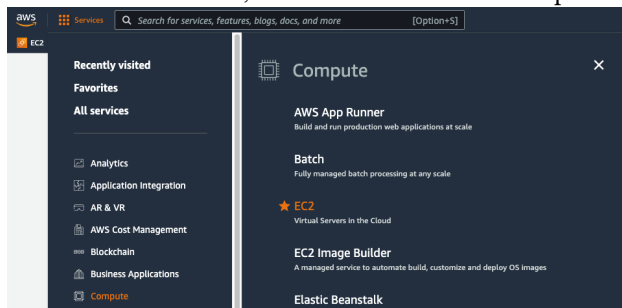
### 1.1 Launching Instances

AWS allows you to create instances (AWS nomenclature for virtual machines) in many different regions in the world. You may go to [https://aws.amazon.com/about-aws/global-infrastructure/regions\\_az/](https://aws.amazon.com/about-aws/global-infrastructure/regions_az/) to check the list of AWS regions around the world. Popular Internet services often have instances running in different regions to ensure that different markets can use the services with low latency.

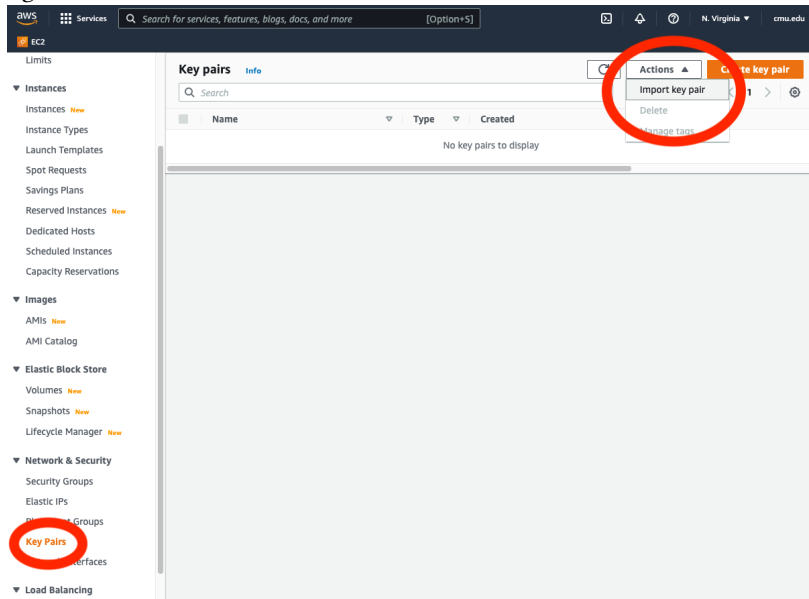
**AWS Console.** To launch an instance go to your AWS console at <https://us-east-1.console.aws.amazon.com/>. Note that we are accessing the console for the us-east-1 region. Every region has a separate AWS console and **you need to repeat the process below for every region where you want to launch an instance.** Here is how the AWS console looks like (note that the region name “N. Virginia” is displayed in the top right corner):



To launch an instance, click on “Services” in the top left corner and select “Compute” and then “EC2.”

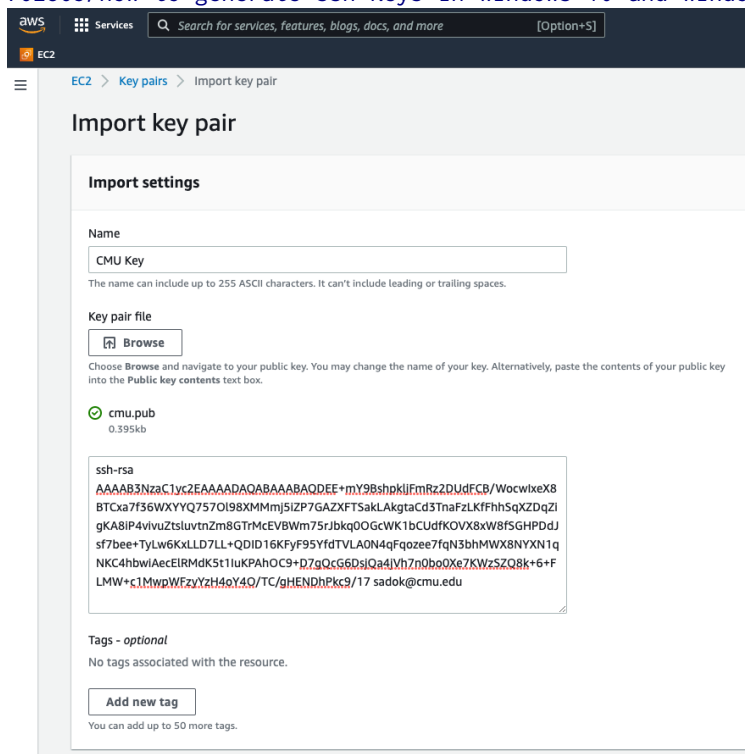


**Key pair.** Before launching an instance, first make sure you have an SSH key set up. Go to “Key Pairs” (under “Network & Security” in the left side bar). You can either click on “Create key pair” to have AWS create an SSH key for you or you can add an existing SSH key by clicking on “Actions” and then “Import key pair” as shown in the next figure.



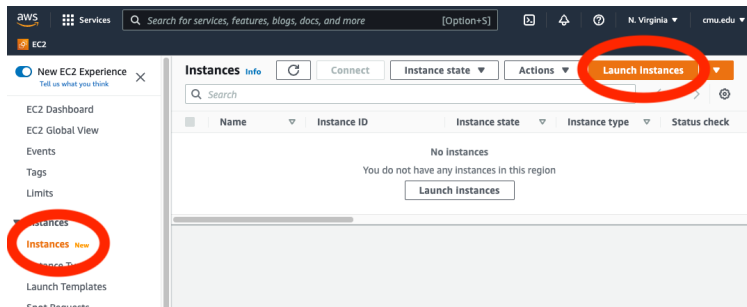
Then, you can either select a file with your public SSH key or paste it in the text box. To finish importing, click on “Import key pair.”

Here is also a brief tutorial on how to generate the key pair on your own laptop: <https://www.howtogeek.com/762863/how-to-generate-ssh-keys-in-windows-10-and-windows-11/>.

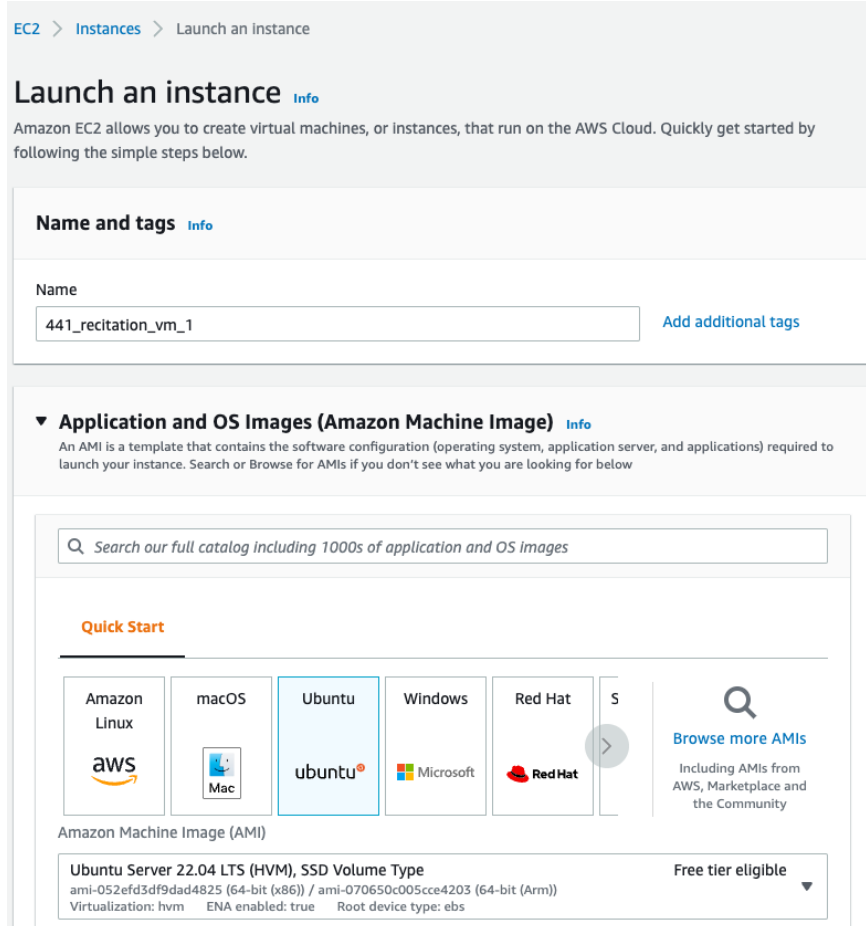


If you create a new key pair, you need to save it to somewhere in your laptop for your future use (later following the connecting to the instance part).

**Launching an instance.** You are now ready to launch instances. Go to the “Instances” panel in the left side panel and click on “Launch instances.”



Give a descriptive name to your instance and select an image. We recommend using “Ubuntu Server 22.04 LTS.”



Select an instance type. The instance type determines the amount of resources that will be available to your instance (e.g., CPU cores and memory). **Please use a free-tier eligible instance (e.g., t2.micro or t2.nano) to prevent overcharging.**

▼ **Instance type** [Info](#)

Instance type

**t2.micro**  
 Family: t2 1 vCPU 1 GiB Memory  
 On-Demand Linux pricing: 0.0116 USD per Hour  
 On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible

▼

[Compare instance types](#)

Select the key that you created in the previous step.

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

▼

[Create new key pair](#)

Now we will configure the VM firewall rules. We are going to create a new “security group” and allow SSH traffic from anywhere (0.0.0.0/0). This new security group will block traffic to all ports but 22 (the port used by SSH), we will open some ports in a later step.

▼ **Network settings** [Info](#) [Edit](#)

Network [Info](#)  
vpc-04a27d21329139ff6

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)  
Enable

**Firewall (security groups)** [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group
 ☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☒ Allow SSH traffic from  
 Helps you connect to your instance
 

Anywhere  
 0.0.0.0/0

☐ Allow HTTPs traffic from the internet  
 To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet  
 To set up an endpoint, for example when creating a web server

⚠ 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.

×

You can leave all the other options unchanged and click on “Launch instance” in the right panel.

Now if you go back to the Instances panel, you should see your instance listed. It may take a little for it to finish booting. You can also see your instance’s public IP. You will use this IP to access your instance using SSH. You may try this now.

Instances (1) <a href="#">Info</a>									
<div> <input type="text" value="Search"/> </div>									
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
<input type="checkbox"/>	441_recitation...	i-03e5161b4ed06282b	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-54-89-215-69.com...	54.89.215.69

**Configuring the security group.** On AWS a security group is responsible for describing the firewall rules that apply to a given instance. Firewall rules determine which traffic is allowed to enter or leave an instance. For example, when creating the instance, we allowed all SSH traffic, therefore the instance's security group should already include rules to allow inbound SSH traffic. If you click your instance and go to the Security panel in the bottom as shown in the next Figure, you will see that it has an inbound rule to allow traffic to port 22.

The screenshot shows the AWS Management Console interface. At the top, there's a header for 'Instances (1/1)' with a search bar and buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below this is a table listing instances. The first instance, '441\_recitation...', is highlighted. Below the table, the details for this instance are shown, including its name, ID, state (Running), type (t2.micro), status check (Initializing), alarm status (No alarms), availability zone (us-east-1a), public IPv4 DNS, public IPv4 address (54.89.215.69), and elastic IP. The 'Security' tab is selected, showing the security group 'sg-0fe31dfc3561d5690 (launch-wizard-1)'. Under 'Inbound rules', a table shows a rule with ID 'sgr-053addf887b2c516f', port range '22', protocol 'TCP', source '0.0.0.0/0', and security groups 'launch-wizard-1'. Under 'Outbound rules', a table shows a rule with ID 'sgr-0565d3b20cb8f0100', port range 'All', protocol 'All', destination '0.0.0.0/0', and security groups 'launch-wizard-1'.

We will now add some extra firewall rules to allow our experiments to go through. Start by clicking on the security group (sg-0fe31dfc3561d5690 (launch-wizard-1) in the example above). You will be directed to the security group page as shown below. Click on “Edit inbound rules.”

The screenshot shows the AWS Management Console interface for the security group 'sg-0fe31dfc3561d5690 - launch-wizard-1'. The 'Details' tab is selected, showing the security group name, ID, description, VPC ID, owner, inbound rules count (1 Permission entry), and outbound rules count (1 Permission entry). Below the details, there are tabs for 'Inbound rules', 'Outbound rules', and 'Tags'. The 'Inbound rules' tab is selected, showing a table with one rule. The rule has ID 'sgr-053addf887b2c516f', IP version 'IPv4', and type 'SSH'. There are buttons for 'Manage tags' and 'Edit inbound rules'.

- Click on “Add rule” to add a new rule to allow iperf3 traffic. Select “Custom TCP” as the Type, “5201” as the

Port range, and “Anywhere-IPv4” as the Source. Note that this will add  $0.0.0.0/0$  as the source, as we will see later in the class this is a CIDR notation to represent all IPv4 addresses.

- Click again on “Add rule” to allow ICMP traffic (we need to allow ICMP in order to allow the instances to be pinged). Select “All ICMP - IPv4” as the Type and “Anywhere-IPv4” as the Source.

After adding these rules, it should look as follows. Click on “Save rules” to apply your changes.

EC2 > Security Groups > sg-0fe31dfc3561d5690 - launch-wizard-1 > Edit inbound rules

### Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	
sgr-053addf887b2c516f	SSH	TCP	22	Custom	<input type="text" value="0.0.0.0/0"/>	Delete
-	Custom TCP	TCP	5201	Anywhere...	<input type="text" value="0.0.0.0/0"/>	Delete
-	All ICMP - IPv4	ICMP	All	Anywhere...	<input type="text" value="0.0.0.0/0"/>	Delete

[Add rule](#)

Cancel [Preview changes](#) [Save rules](#)

**Terminating an instance.** Make sure to terminate your instance after you are done. Otherwise the instance will continue to run and drain your credits. To do so, you may right click the instance and select “Terminate instance.” When prompted, confirm that you want to terminate the instance.

Instances (1/1) [Info](#) [Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input checked="" type="checkbox"/>	441_recitation...	i-03e5161b4ed06282b	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-89-215-69.com...	54.89.215.69	-

Instance: i-03e5161b4ed06282b (441\_recitation\_vm\_1)

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

**Instance summary [Info](#)**

Instance ID i-03e5161b4ed06282b (441_recitation_vm_1)	Public IPv4 address 54.89.215.69   <a href="#">open address</a>	Private IPv4 addresses 172.31.94.157
--	--	---

Launch instances  
Launch instance from template  
Migrate a server  
Connect  
Stop instance  
Start instance  
Reboot instance  
Hibernate instance  
**Terminate instance**  
Instance settings  
Networking  
Security  
Image and templates  
Monitor and troubleshoot

## 2 Running Experiments

### 2.1 Connecting to the instance.

**Find the public IP address for your instance.** On the same page from the last step, click "Connect":

EC2 > Instances > i-0671ff0a4debe6acc

**Instance summary for i-0671ff0a4debe6acc (test-korea)** Info

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state](#) [Actions](#)

Instance ID i-0671ff0a4debe6acc (test-korea)	Public IPv4 address 43.200.179.222   <a href="#">open address</a>	Private IPv4 addresses 172.31.33.110
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-43-200-179-222.ap-northeast-2.compute.amazonaws.com   <a href="#">open address</a>
Hostname type IP name: ip-172-31-33-110.ap-northeast-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-33-110.ap-northeast-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.   <a href="#">Learn more</a>
Auto-assigned IP address 43.200.179.222 [Public IP]	VPC ID vpc-047d444ae226c5a74	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-03e82e63224209892	

Under the tab of **SSH Client**, find the command to connect to your instance:

EC2 > Instances > i-0671ff0a4debe6acc > Connect to instance

**Connect to instance** Info

Connect to your instance i-0671ff0a4debe6acc (test-korea) using any of these options

[EC2 Instance Connect](#) | [Session Manager](#) | [SSH client](#) | [EC2 serial console](#)

Instance ID  
i-0671ff0a4debe6acc (test-korea)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is thinkpad.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 thinkpad.pem
4. Connect to your instance using its Public DNS:  
ec2-43-200-179-222.ap-northeast-2.compute.amazonaws.com

Example:  
ssh -i "thinkpad.pem" ubuntu@ec2-43-200-179-222.ap-northeast-2.compute.amazonaws.com

**Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

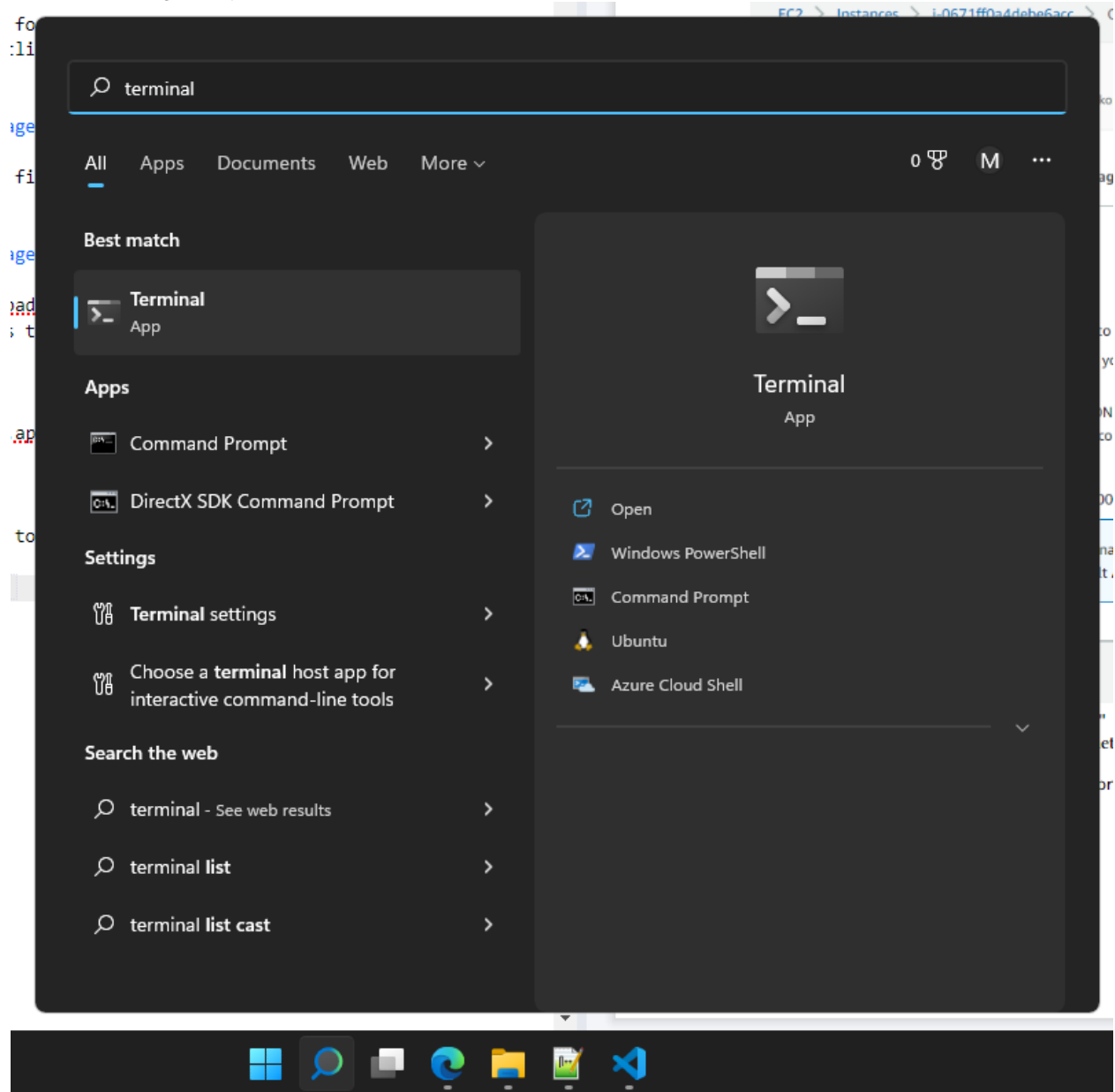
[Cancel](#)

The key parameter `-i "thinkpad.pem"` is the file that you saved in the key pair step. If you upload your own keys to AWS, you may not need the parameter, so the command may be like:

```
ssh ubuntu@ec2-43-200-179-222.ap-northeast-2.compute.amazonaws.com
```

**Connect to your instance.** Now you need to use your own terminal to connect to the AWS instance. First you need to find the terminal on your own laptop. For example, for Windows users, we recommend you to use the Windows

terminal, coming with your own Windows OS.



Then, type the command starting with `ssh` you copy from the last step. You will have the access to your own instance now!



```
ubuntu@ip-172-31-33-110: ~  
# Zili Meng @ zili-thinkpad in ~ [15:50:07]  
$ ssh ubuntu@ec2-43-200-179-222.ap-northeast-2.compute.amazonaws.com  
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1011-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Thu Sep  1 19:50:19 UTC 2022  
  
System load:  0.0      Processes:            99  
Usage of /:   29.5% of 7.58GB  Users logged in:     0  
Memory usage: 25%      IPv4 address for eth0: 172.31.33.110  
Swap usage:   0%  
  
* Ubuntu Pro delivers the most comprehensive open source security and  
  compliance features.  
  
https://ubuntu.com/aws/pro  
  
44 updates can be applied immediately.  
7 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
*** System restart required ***  
Last login: Wed Aug 31 20:24:52 2022 from 128.237.82.13  
ubuntu@ip-172-31-33-110:~$ |
```

Note that for P0, you may need to create multiple instances.

## 2.2 Using ping

Using ping is relatively easy. ping comes with the operating system, so just type ping and the IP address or the domain that you want to ping, like:

```
ping google.com  
ping 8.8.8.8
```

Now you will see the round-trip time (RTT) from the instance where you initiate the ping and the host that you are pinging.

```
ubuntu@ip-172-31-33-110:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=105 time=30.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=105 time=30.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=105 time=30.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=105 time=30.4 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=105 time=30.5 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=105 time=30.5 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=105 time=30.4 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=105 time=30.5 ms
^C
--- 8.8.8.8 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7012ms
rtt min/avg/max/mdev = 30.422/30.469/30.523/0.029 ms
```

From the results, you can see that your instance will try to ping the host every second. When you terminate the program, it will report the average and some other statistics of the RTTs. You can try to explore other usages by running `ping --help`.

Note that default ping on Ubuntu will not terminate unless you manually terminate it with Ctrl+C.

## 2.3 Using iperf3

iperf3 requires you to run commands from both sides. If you do not yet have two instances, repeat the process in §1.1 to launch another instance.

On each instance, first you need to install iperf3 by:

```
sudo apt update && sudo apt install iperf3
```

In the following you need to have different actions on two instances. On **Instance 1**, run:

```
iperf3 -s
```

This starts an iperf3 server.

On **Instance 2**, run:

```
iperf3 -c <server-ip>
```

This starts an iperf3 client. <server-ip> is the public IP address of the **Instance 1**.

```

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-33-110:~$ iperf3 -s
-----
Server listening on 5201
Accepted connection from 52.67.96.220, port 60998
[ 5] local 172.31.33.110 port 5201 connected to 52.67.96.220 port 32768
[ ID] Interval      Transfer    Bitrate
[ 5] 0.00-1.00 sec  42.4 KBytes 347 Kbits/sec
[ 5] 1.00-2.00 sec  464 KBytes 3.80 Mbits/sec
[ 5] 2.00-3.00 sec  2.96 MBytes 24.8 Mbits/sec
[ 5] 3.00-4.00 sec  5.68 MBytes 47.7 Mbits/sec
[ 5] 4.00-5.00 sec  3.29 MBytes 27.6 Mbits/sec
[ 5] 5.00-6.00 sec  3.17 MBytes 26.6 Mbits/sec
[ 5] 6.00-7.00 sec  5.98 MBytes 50.2 Mbits/sec
[ 5] 7.00-8.00 sec  4.85 MBytes 40.7 Mbits/sec
[ 5] 8.00-9.00 sec  5.68 MBytes 47.7 Mbits/sec
[ 5] 9.00-10.00 sec 4.82 MBytes 40.4 Mbits/sec
[ 5] 10.00-10.28 sec 1.50 MBytes 44.5 Mbits/sec
-----
[ ID] Interval      Transfer    Bitrate
[ 5] 0.00-10.28 sec 38.4 MBytes 31.3 Mbits/sec
-----
Server listening on 5201

Last login: Wed Aug 31 20:24:02 2022 from 128.237.82.13
ubuntu@ip-172-31-2-18:~$ iperf3 -c 43.200.179.222
Connecting to host 43.200.179.222, port 5201
[ 5] local 172.31.2.18 port 32768 connected to 43.200.179.222 port 5201
[ ID] Interval      Transfer    Bitrate    Retr    Cwnd
[ 5] 0.00-1.00 sec  365 KBytes  2.99 Mbits/sec  0    56.6 KBytes
[ 5] 1.00-2.00 sec  2.75 MBytes 23.0 Mbits/sec  0    477 KBytes
[ 5] 2.00-3.00 sec  3.75 MBytes 31.5 Mbits/sec  0    3.01 MBytes
[ 5] 3.00-4.00 sec  5.00 MBytes 41.9 Mbits/sec  0    3.01 MBytes
[ 5] 4.00-5.00 sec  3.75 MBytes 31.4 Mbits/sec  159   863 KBytes
[ 5] 5.00-6.00 sec  2.50 MBytes 21.0 Mbits/sec  2    1.52 MBytes
[ 5] 6.00-7.00 sec  6.25 MBytes 52.4 Mbits/sec  0    1.52 MBytes
[ 5] 7.00-8.00 sec  5.00 MBytes 41.9 Mbits/sec  0    1.52 MBytes
[ 5] 8.00-9.00 sec  6.25 MBytes 52.4 Mbits/sec  0    1.52 MBytes
[ 5] 9.00-10.00 sec  3.75 MBytes 31.5 Mbits/sec  0    1.52 MBytes
-----
[ ID] Interval      Transfer    Bitrate    Retr
[ 5] 0.00-10.00 sec 39.4 MBytes 33.0 Mbits/sec  161
[ 5] 0.00-10.28 sec 38.4 MBytes 31.3 Mbits/sec
-----
iperf Done.
ubuntu@ip-172-31-2-18:~$

```

You've got it! Now you can see the bandwidth measurement results.

You can try to explore other usages by running `iperf3 --help` (e.g., for UDP traffic).

Now try on the P0! You almost have already had everything set up for P0.