Computer Security Fundamentals – Spring 2023

Security in Cloud Computing Environments (AWS) Total: 10 points

[Pre-Required knowledge] Please reads this carefully! We will need to use this later!

1) SSH client (Putty in Windows, or? in macOS)

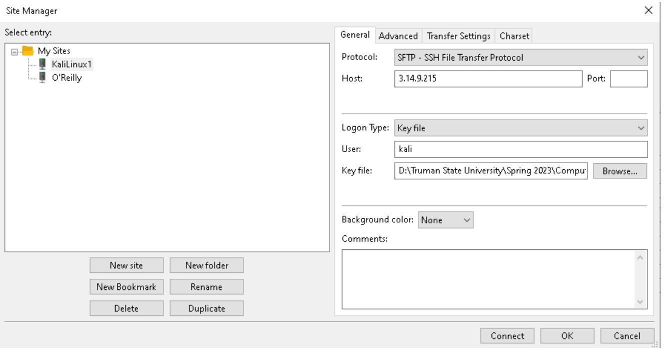
In this homework, you are required to know how to use SSH to connect to the EC2 instance. In my slides, I use Windows as my demonstration machine with Putty as client. I use Putty to load the "private key" which is generated during the process of ECS instance creation.

If you are using some other systems, i.e. macOS, you need to figure out how to use macOS specific SSH clients. Also, you need to figure out how to load the private key to help you to connect to the EC2 instance. My key file, for example, KaliLinuxAWS.ppk is generated in the middle of the EC2 instances creation. "ppk" format is specifically for "Putty in Windows"

About the way to load the key in Putty (in Putty, Windows), please check my slides

2) Secure FTP Client (FileZilla in Windows, or? in macOS)

There is a tool called FileZilla in Windows, what I'm going to do is to upload my KaliLinuxAWS.ppk from my Windows to Kali Linux EC2 instance (I choose key the file just downloaded in the process of key creation!)



Protocol: SFTP - SSH File Transfer Protocol

Host: 3.14.9.215 Port:

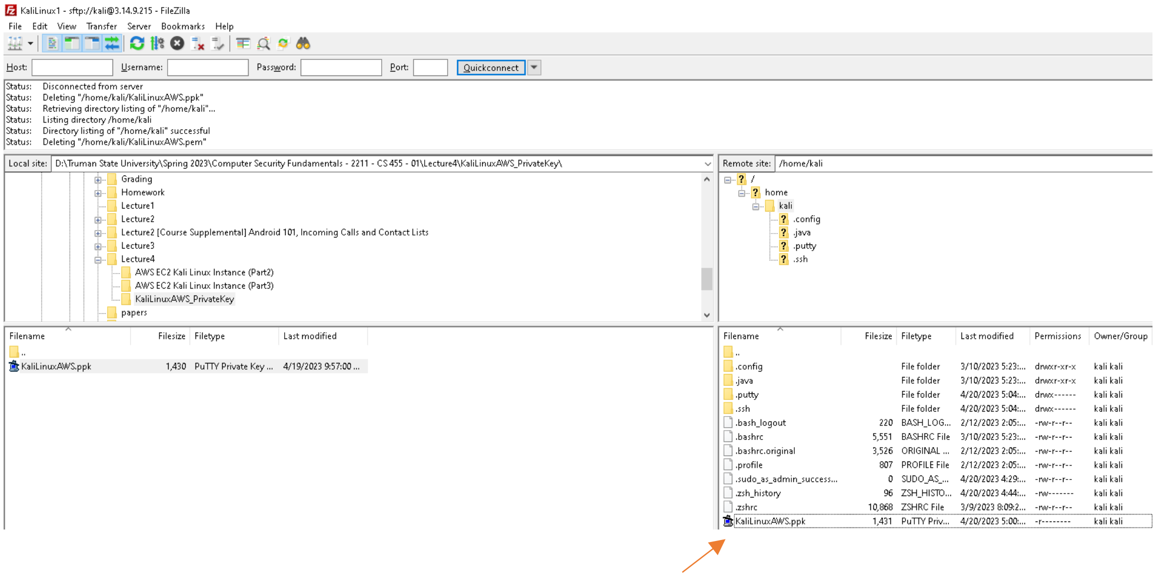
Logon Type: Key file

User: kali

Key file: D:\Truman State University\Spring 2023\Compu

Background color: None

After a successful connection, I just drag-and-drop! Done!


KaliLinux1 - sftp://kali@3.14.9.215 - FileZilla
File
Edit
View
Transfer
Server
Bookmarks
Help
Host:
Username:
Password:
Port:
Quickconnect
Grading
Homework
Lecture1
Lecture2
Lecture2 [Course Supplemental] Android 101, Incoming Calls and Contact Lists
Lecture3
Lecture4
AWS EC2 Kali Linux Instance (Part2)
AWS EC2 Kali Linux Instance (Part3)
KaliLinuxAWS_PrivateKey
papers
/
home
kali
.config
.java
.putty
.ssh

```
Status:   Disconnected from server
Status:   Deleting "/home/kali/KaliLinuxAWS.ppk"
Status:   Retrieving directory listing of "/home/kali"...
Status:   Listing directory /home/kali
Status:   Directory listing of "/home/kali" successful
Status:   Deleting "/home/kali/KaliLinuxAWS.pem"
```

Local site: D:\Truman State University\Spring 2023\Computer Security Fundamentals - 2211 - CS 455 - 01\Lecture4\KaliLinuxAWS_PrivateKey\

Filename	Filesize	Filetype	Last modified
..			
KaliLinuxAWS.ppk	1,430	PuTTY Private Key ...	4/19/2023 9:57:00 ...

Remote site: /home/kali

Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
.config		File folder	3/10/2023 5:23:...	drwxr-xr-x	kali kali
.java		File folder	3/10/2023 5:23:...	drwxr-xr-x	kali kali
.putty		File folder	4/20/2023 5:04:...	drwx------	kali kali
.ssh		File folder	4/20/2023 5:04:...	drwx------	kali kali
.bash_logout	220	BASH_LOG...	2/12/2023 2:05:...	-rw-r--r--	kali kali
.bashrc	5,551	BASHRC File	3/10/2023 5:23:...	-rw-r--r--	kali kali
.bashrc.original	3,526	ORIGINAL ...	2/12/2023 2:05:...	-rw-r--r--	kali kali
.profile	807	PROFILE File	2/12/2023 2:05:...	-rw-r--r--	kali kali
.sudo_as_admin_success...	0	SUDO_AS_...	4/20/2023 4:29:...	-rw-r--r--	kali kali
.zsh_history	96	ZSH_HISTO...	4/20/2023 4:44:...	-rw-------	kali kali
.zshrc	10,868	ZSHRC File	3/9/2023 8:09:2...	-rw-r--r--	kali kali
KaliLinuxAWS.ppk	1,431	PuTTY Priv...	4/20/2023 5:00:...	-r--------	kali kali

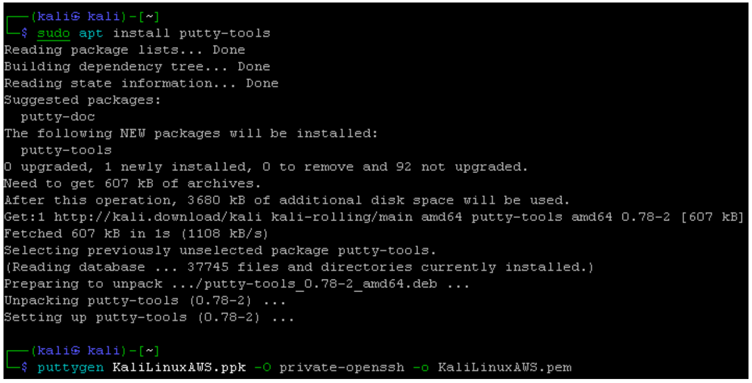
3) putty-tools in Kali Linux for key format transformation

- **Installation**

sudo apt install putty-tools

- **Transform**

puttygen KaliLinuxAWS.ppk -O private-openssh -o KaliLinuxAWS.pem



```
┌──(kali㉿kali)-[~]
└─$ sudo apt install putty-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  putty-doc
The following NEW packages will be installed:
  putty-tools
0 upgraded, 1 newly installed, 0 to remove and 92 not upgraded.
Need to get 607 kB of archives.
After this operation, 3680 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 putty-tools amd64 0.78-2 [607 kB]
Fetched 607 kB in 1s (1108 kB/s)
Selecting previously unselected package putty-tools.
(Reading database ... 37745 files and directories currently installed.)
Preparing to unpack .../putty-tools_0.78-2_amd64.deb ...
Unpacking putty-tools (0.78-2) ...
Setting up putty-tools (0.78-2) ...

┌──(kali㉿kali)-[~]
└─$ puttygen KaliLinuxAWS.ppk -O private-openssh -o KaliLinuxAWS.pem
```

The reason we are going to do key format transformation is that.
In Windows, we use Putty for .ppk file, but in the KaliLinux, if we use ssh command, it needs OpenSSH's format, the .pem format!

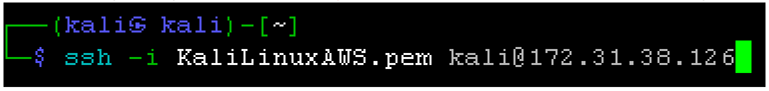
4) Change the access rights

sudo chmod 400 KaliLinuxAWS.pem

Since we now have both .ppk and .pem file, now we need to test if we can use ssh command to make a connection from **KaliLinux1 → KaliLinux2**

If we do not change the access rights, it will give you the error like: access right is too open and it will refuse your connection.

We can specify KaliLinux2's private IP address because they are now in the same AWS cloud



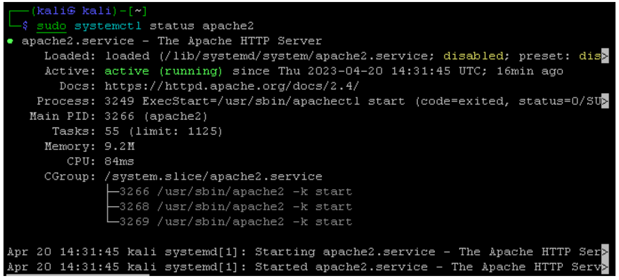
```
┌──(kali㉿kali)-[~]
└─$ ssh -i KaliLinuxAWS.pem kali@172.31.38.126
```

5) Apache Web Server (this will be setup to your KaliLinux2)

In this homework, you are required to know how to install apache web server in EC2 instance (Kali Linux).
Even though I never demonstrate this in the class, it is very easy. After your logging into the EC2 instance, here are the commands you need.

- **sudo apt update**
  - **Try to update (sync) your linux's package with several repository servers**
- **sudo apt install apache2**
  - **This will install everything apache2 by default (version 2.x.x)**
- **You can start apache2 quickly, here is the tutorial https://www.cyberciti.biz/faq/star-stop-restart-apache2-webserver/**
  - **I just use: sudo service apache2 restart**
- **Check the status if it is really started?**

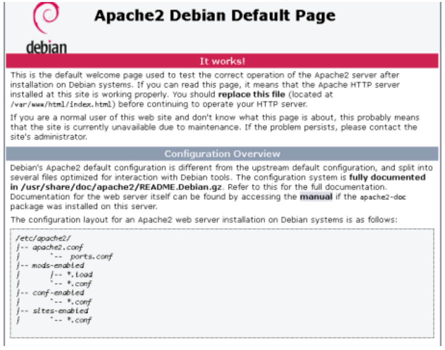
    - **sudo systemctl status apache2**
    - **If you see the message like this, you make it!**



```
┌──(kali㉿ kali)-[~]
└─$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: dis>
     Active: active (running) since Thu 2023-04-20 14:31:45 UTC; 16min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 3249 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
   Main PID: 3266 (apache2)
      Tasks: 55 (limit: 1125)
     Memory: 9.2M
        CPU: 84ms
     CGroup: /system.slice/apache2.service
             ├─3266 /usr/sbin/apache2 -k start
             ├─3268 /usr/sbin/apache2 -k start
             └─3269 /usr/sbin/apache2 -k start

Apr 20 14:31:45 kali systemd[1]: Starting apache2.service - The Apache HTTP Ser>
Apr 20 14:31:45 kali systemd[1]: Started apache2.service - The Apache HTTP Serv>
```

- **See if you can see this page? But you need to do some works in the [Security Groups] first!**



debian

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

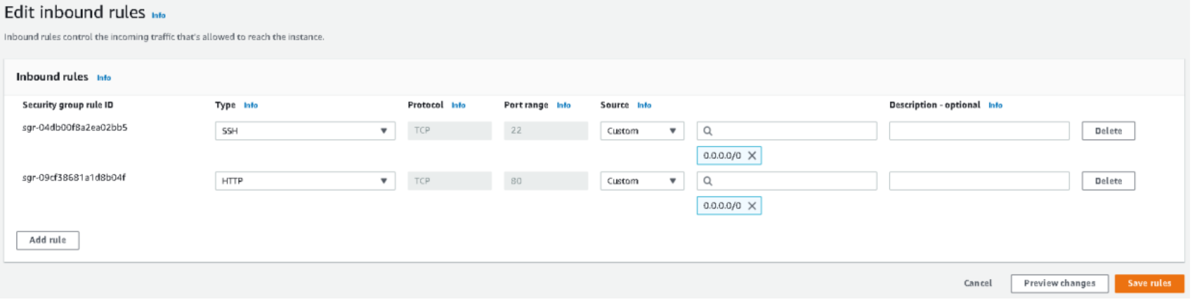
Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

    - **Go to the [Security Groups] for this (KaliLinux2) EC2 instance and do the following setting. The part of the connection sources could be 0.0.0.0/0, or from your IP Address. Check the following as an example of my configuration. Then, I believe, with a correct setting, you can see the "Apache2 Debian Default Page".**



Edit inbound rules Info

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

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-04db00f8a2ea02bb5	SSH ▼	TCP	22	Custom ▼	0.0.0.0/0 ✕		Delete
sgr-09cf38681a1d8b04f	HTTP ▼	TCP	80	Custom ▼	0.0.0.0/0 ✕		Delete

    - **Of course, you need to know your instance's IP Address (IPv4).**
        - **Check my slide for knowing IP address in the EC2 instance**
    - **Then, you can open your browser and type the IP address. You can see the webpage.**

6) **Text Editor (vim) in Kali Linux**

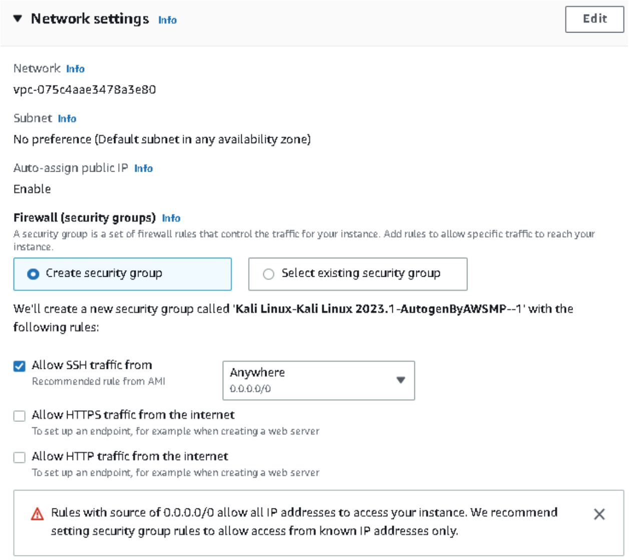
   You might need to install text editor (vim) **in the command line environment** in editing your **html code**. If you know how to use this would be wonderful. You can easily Google that about its instructions or tutorials.

   sudo apt install vim

[Steps]

1) **Create an AWS account, follow my instructions in the course slides**
2) **Get into the EC2 services and create 2 * EC2 instances.**
    - **Remember to setup BOTH as t2.micro (with free tier support), otherwise you will get charged with some trivial amount of $**

- **Follow my slides to create the key pair (public + private). Private key will be downloaded and if you are using Putty, please choose the correct format!**
- **For the 2nd EC2 instance, you can reuse the same key file, or to generate a new one. (I prefer to share the same key file)**
- **Keep all of the Network settings by its default. If you want to put something, it is OK. You need to be VERY SURE it can work!**



▼ Network settings Info

Network Info
vpc-075c4aae3478a3e80

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 '**Kali Linux-Kali Linux 2023.1-AutogenByAWSMP--1**' with the following rules:

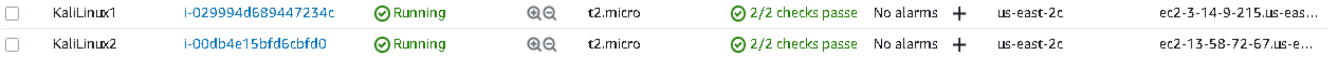
☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Recommended rule from AMI

□ 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.

- **For the storage, just follow the default setting to give both of the instances 12GB with gp2 format. 12 + 12 would be ok. If it is over 30GB, you might get charged with $.**
- **Once you finished the setup of 2 instances, you can see something like this.**
  I use KaliLinux1 and KaliLinux2 as their names in the process of instance creation

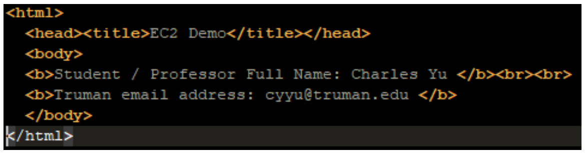


□	KaliLinux1	i-029994d689447234c	✓ Running	t2.micro	✓ 2/2 checks passe	No alarms	+	us-east-2c	ec2-3-14-9-215.us-eas...
□	KaliLinux2	i-00db4e15bfd6cbfd0	✓ Running	t2.micro	✓ 2/2 checks passe	No alarms	+	us-east-2c	ec2-13-58-72-67.us-e...

- **So now, once you finished the basic setting of 2 instances, you need to try to connect to 2 instances by using SSH (because it is allowed in the Security Group, by default) with your private key from your local laptop / computer. I use Putty as my SSH client**
- **Follow my slides, try to put down IP addresses (public + private) for your 2 * EC2 instances**
- **You are almost done now! The 'basic instances creation' is done!**

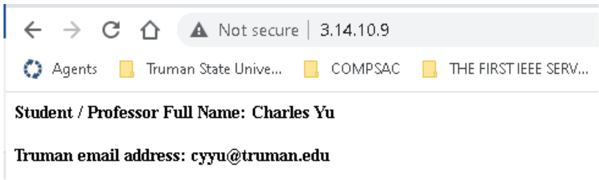
3) Follow the [Pre-Required Knowledge], use the SSH, go into the KaliLinux2 to setup the following

- **An Apache web server. Then, start the http service**
- **Check if the http service is successfully started?**
- **Go to its (instance2, in my case, it is KaliLinux2) folder, /var/www/html and rename the index.html as index.html.old**
  - **Now new create a webpage as follows, you need to replace my email and my name as yours**



```
<html>
  <head><title>EC2 Demo</title></head>
  <body>
  <b>Student / Professor Full Name: Charles Yu </b><br><br>
  <b>Truman email address: cyyu@truman.edu </b>
  </body>
</html>
```

- **Note: You need to use “sudo” to get the root’s access rights in this folder in dong the jobs like editing or files creation.**
- **If it is working, go back to the browser. You need to input your Instance2’s Public IP address in the address line. Hit the [Enter] and see if it is working?**



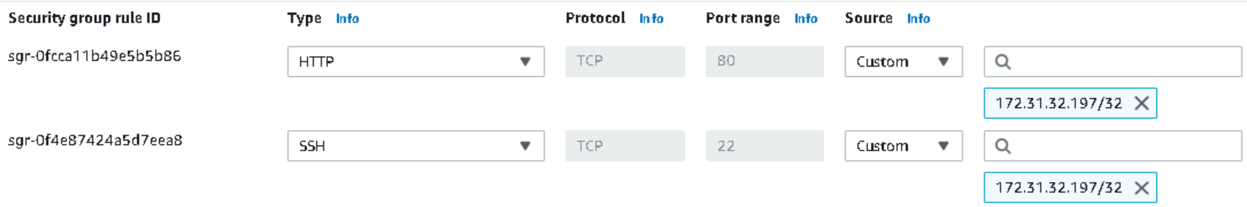
Student / Professor Full Name: Charles Yu

Truman email address: cyyu@truman.edu

- **Please carefully check your html syntax. If your page is not rendered correctly, most of the time, it is the “typo” of your html syntax**

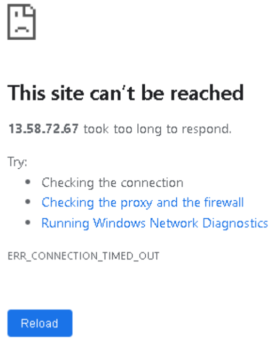
4) Now, we want to do experiments to limit our access rights (our machine, not the EC2 instances)

- Click the KaliLinux2 (your 2nd instance) in the ‘Instances’
- Click the ‘Security’ tab
- Go to its ‘Security Groups’
- Click the [Edit inbound rules]
- Limit the SSH and HTTP traffic from our KaliLinux1 only, by inputting its private IP Address!



Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	
sgr-0fcca11b49e5b5b86	HTTP ▼	TCP	80	Custom ▼	172.31.32.197/32 ✕
sgr-0f4e87424a5d7eea8	SSH ▼	TCP	22	Custom ▼	172.31.32.197/32 ✕

- So, in this way, I cannot connect KaliLinux2 and visit its webpage from my laptop



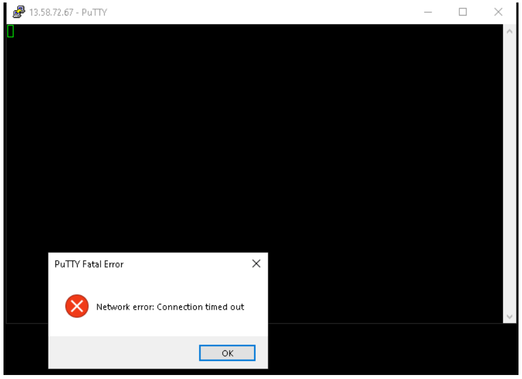
This site can’t be reached

13.58.72.67 took too long to respond.

Try:
- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

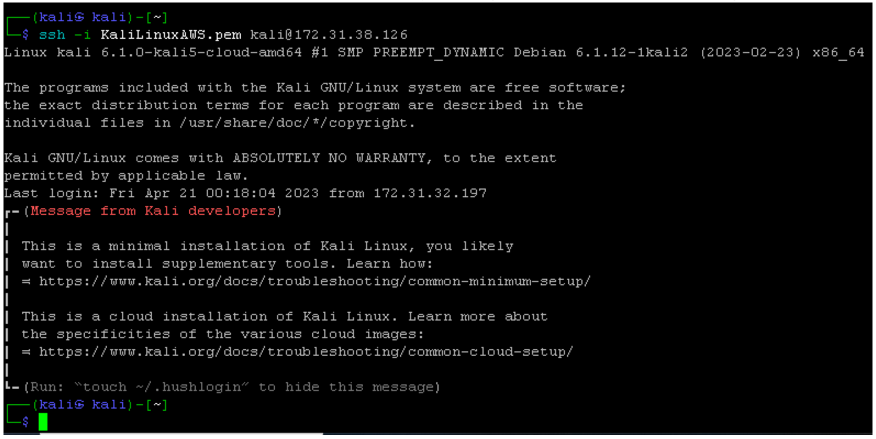
ERR_CONNECTION_TIMED_OUT

- Also, I cannot connect to KaliLinux2 by using SSH from my laptop (timeout)



Network error: Connection timed out

- Theoretically, we can view KaliLinux2's webpage from KaliLinux1 but we don't have the browser.
- There is one more thing we can do. We can login to KaliLinux1 by using SSH. Then? Login to KaliLinux2 from KaliLinux1 by using SSH. This is working!



```
┌──(kali㉿ kali)-[~]
└─$ ssh -i KaliLinuxAWS.pem kali@172.31.38.126
Linux kali 6.1.0-kali5-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.12-1kali2 (2023-02-23) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 21 00:18:04 2023 from 172.31.32.197
┏━(Message from Kali developers)
┃
┃ This is a minimal installation of Kali Linux, you likely
┃ want to install supplementary tools. Learn how:
┃ ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
┃
┃ This is a cloud installation of Kali Linux. Learn more about
┃ the specificities of the various cloud images:
┃ ⇒ https://www.kali.org/docs/troubleshooting/common-cloud-setup/
┃
┗━(Run: "touch ~/.hushlogin" to hide this message)
┌──(kali㉿ kali)-[~]
└─$ █
```

[Requirements] You can put everything into Word or PDF file! For submission, I only need 1 file. No programming at all in this homework! (HTML code is provided)

Q1. I want 1 screenshot for 2 instances, "Instance state" has to be "Running" (1 pt)

Q2. I want all the information, as follows, your (2 pts)

- Public and Private IP Address in Instance1
- Public and Private IP Address in Instance2

Q3. I want 1 screenshot for your folder, in your Instance1, after the key conversion by using the "puttygen" command. So, both of the keys, .ppk and .pem files has to be there in the picture! (1 pt)

Q4. I want 1 screenshot for your webpage, which is saying your Apache server is working in your Instance2. In this webpage, it has your full name and your Truman email. This is like what I have done earlier. (1 pt)

Q5. I want 1 screenshot for your Instance2’s “Security Group” setting (inbound rule). You need to limit the connection **only** from Instance1. You need to put Instance1’s private IP address
(1 pt)

Q6. I want 1 screenshot to show you are unable to open the webpage in the browser to the Instance2 anymore (from your laptop) (Instance2 has apache web server configured previously)
(1pt)

Q7. I want 1 screenshot to show you are unable to connect to the Instance2 by using SSH (from your laptop) anymore. There will be a time out error message popped up (1 pt)

Q8. I want 1 screenshot to show you are able to connect to the Instance2 by using SSH from Instance1, like I had done previously. (You need to specify the .pem key) (1 pt)

Q9. Finally, I want two screenshots for your storage in 2 instances.
You can go to Instances. Check the instance. Then, check the “Storage”
Scroll down. You will see the “Volumn ID” as well as “Volumn Size (GiB)”
I want to make sure that, storage (Instance1) + storage (Instance2) <= 30 GB.
Otherwise, you will get charged with some $ (1 pt)