

Ocean Lu  
Professor Lan Yang  
CS 4650  
10/16/2019

## Activity 4: EC2 Report

### 1. Launching instance:

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile. The left sidebar contains a navigation menu with categories like EC2 Dashboard, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area is titled 'Resources' and shows a summary of EC2 resources in the US East (N. Virginia) region. A 'Create Instance' button is prominently displayed. The 'Service Health' section indicates that the service is operational. The right sidebar contains 'Account Attributes' and 'Additional Information' links.

### 2. Chose Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0b69ea66ff7391e80 (64-bit x86) / ami-09c61c4850b7465cb (64-bit Arm)

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' screen in the AWS Management Console. The screen has a progress bar at the top indicating the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The main content area shows a search bar and a list of AMIs. The 'Quick Start' section highlights the 'Amazon Linux 2 AMI (HVM), SSD Volume Type' with the ID 'ami-0b69ea66ff7391e80 (64-bit x86) / ami-09c61c4850b7465cb (64-bit Arm)'. Other AMIs like 'Amazon Linux AMI 2018.03.0' and 'Red Hat Enterprise Linux 8' are also visible. The bottom of the screen shows the 'Feedback' and 'English (US)' options.

### 3. Choose a tier

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn 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)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

### 4. Configure instances

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

**Number of instances** 1 [Launch into Auto Scaling Group](#)

**Purchasing option** ☐ Request Spot instances

**Network** vpc-b7d6b8cd (default) [Create new VPC](#)

**Subnet** No preference (default subnet in any Availability Zone) [Create new subnet](#)

**Auto-assign Public IP** Use subnet setting (Enable)

**Placement group** ☐ Add instance to placement group

**Capacity Reservation** Open [Create new Capacity Reservation](#)

**IAM role** None [Create new IAM role](#)

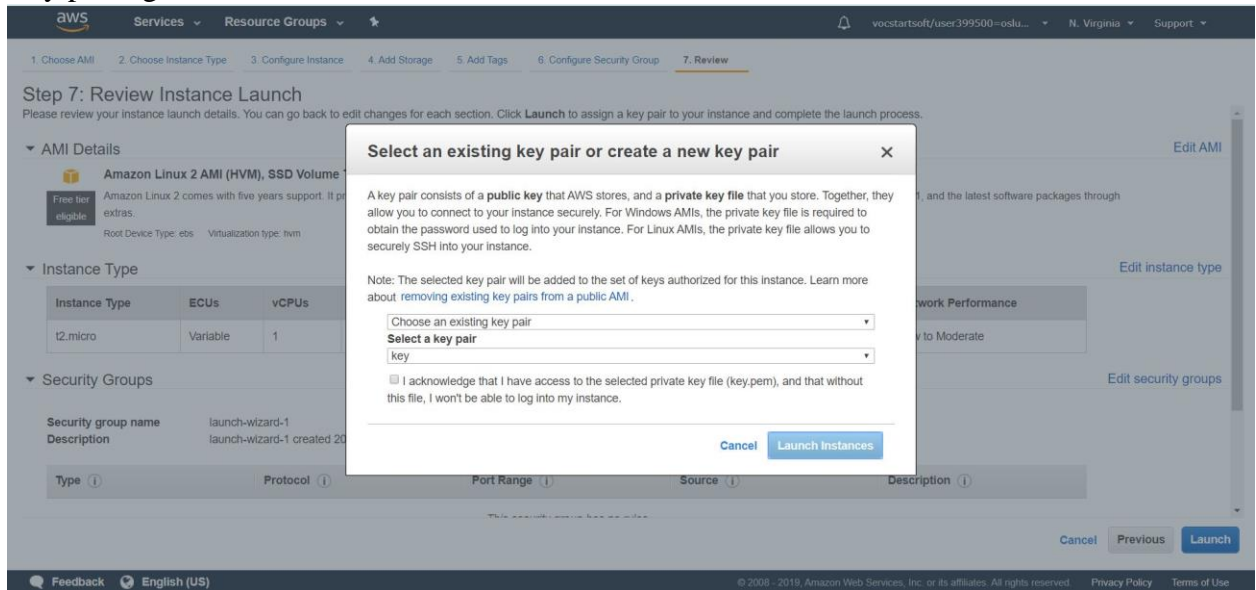
**Shutdown behavior** Stop

**Enable termination protection** ☐ Protect against accidental termination

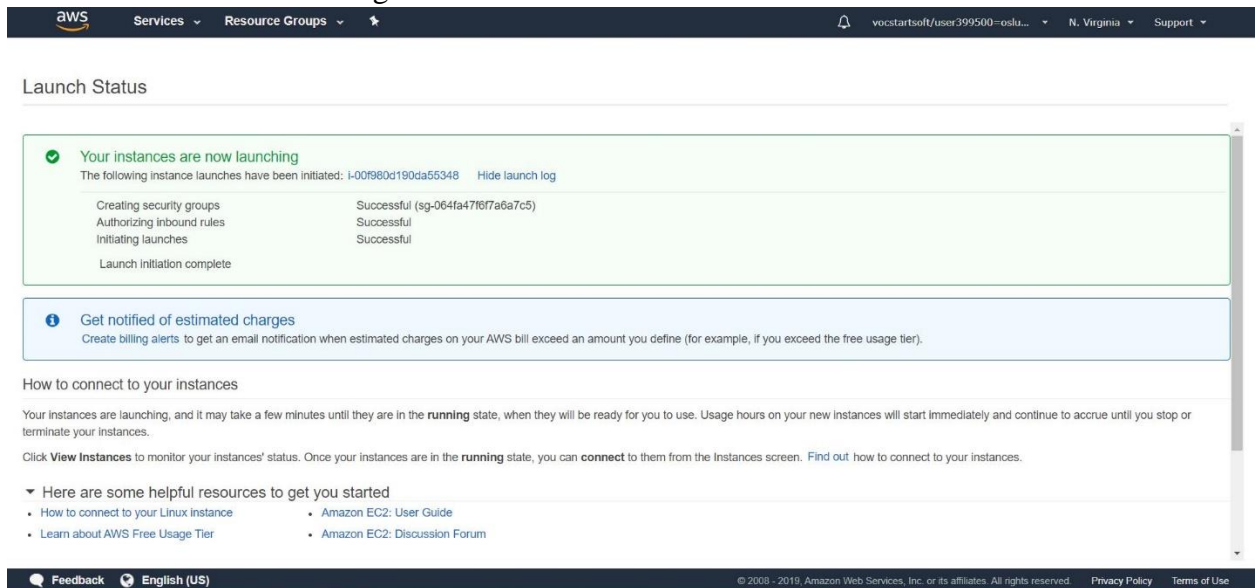
**Monitoring** ☐ Enable CloudWatch detailed monitoring  
Additional charges apply.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

## 5. Key pairing



## 6. Launch instance and view log



## 7. Connect to SSH

The screenshot displays the AWS Management Console interface. On the left, the 'Amazon Elastic Compute Cloud' sidebar is visible with a search bar and a list of navigation links including 'What is Amazon EC2?', 'Setting Up', 'Getting Started', 'Best Practices', 'Tutorials', 'Amazon Machine Images', 'Instances', 'Instance Types', 'Instance Purchasing Options', 'Instance Lifecycle', 'Launch', 'Connect', 'Prerequisites for Connecting', and 'Connect Using SSH'. The main content area is titled 'Connect to Your Linux Instance using an SSH Client' and provides instructions on how to connect to a Linux instance using an SSH client. It includes a terminal snippet for the SSH command: `ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com`. Below this, it shows a response from the host: 'The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established. RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f. Are you sure you want to continue connecting (yes/no)?'. The second step explains how to connect using an IPv6 address, with a terminal snippet: `ssh -i /path/my-key-pair.pem ec2-user@2001:db8:1234:1a00:9691:9503:25ad:1761`. On the right, a 'Prerequisites' section lists 'Connect to Your Linux Instance using an SSH Client' and 'Transferring Files to Linux Instances from Linux Using SCP'. At the bottom, a modal window titled 'Connect To Your Instance' is open, showing options to connect via a standalone SSH client, EC2 Instance Connect, or a Java SSH Client. It provides steps to access the instance, including opening an SSH client, locating the private key file, setting permissions with `chmod 400 key.pem`, and connecting using the Public DNS: `ec2-3-83-115-230.compute-1.amazonaws.com`. An example command is shown: `ssh -i "key.pem" ec2-user@ec2-3-83-115-230.compute-1.amazonaws.com`. The modal also includes a note about the username and a link to connection documentation.

My Script: `ssh -i "key.pem" ec2-user@ec2-3-83-115-230.compute-1.amazonaws.com`

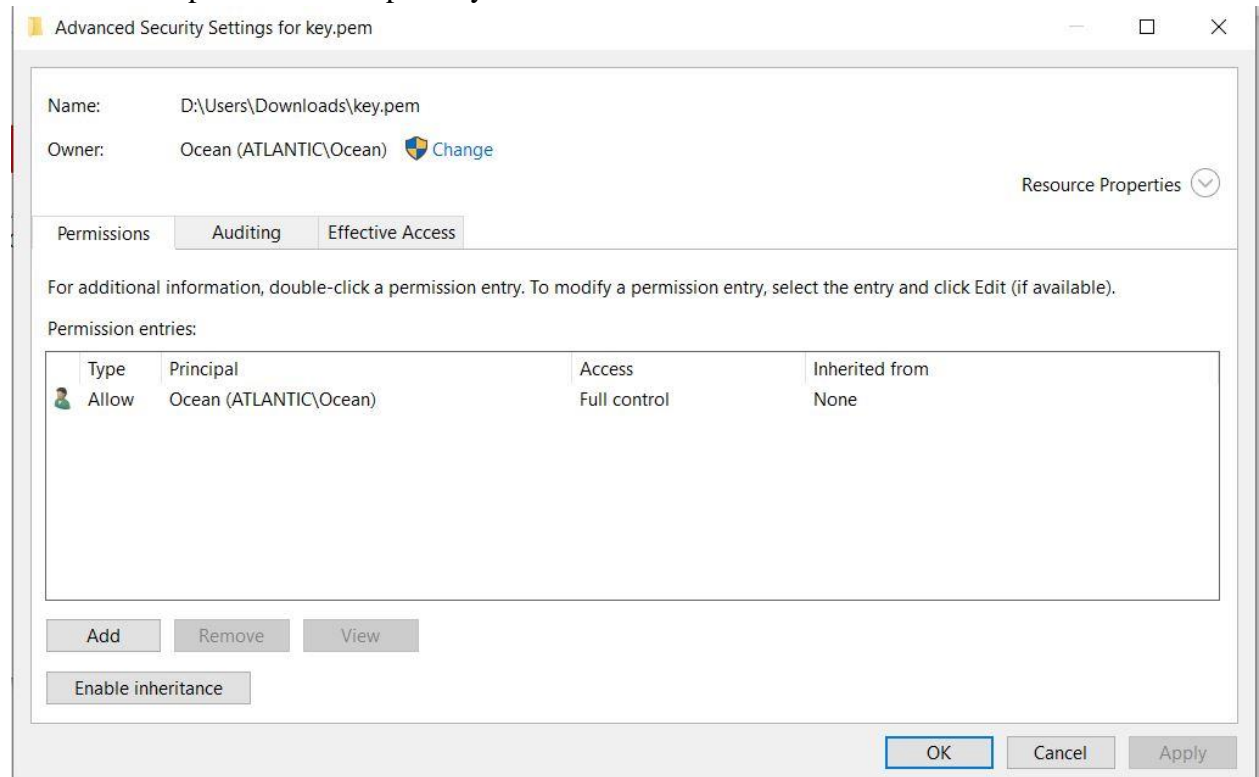
For windows, it is important to change the permission on the private key file

a. Error message:

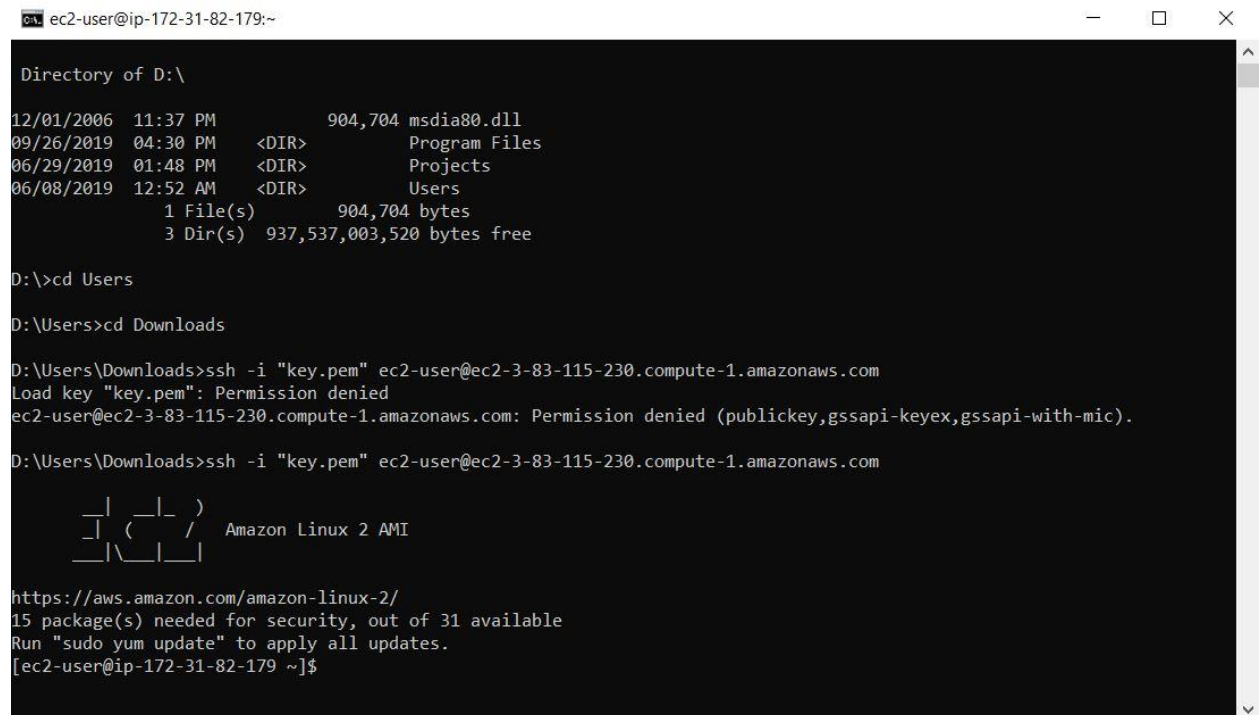
```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions for 'key.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "key.pem": bad permissions
ubuntu@192.168.0.1: Permission denied (publickey).
```

b. Locate the file in Windows Explorer, right-click on it then select "Properties". Navigate to the "Security" tab and click "Advanced".

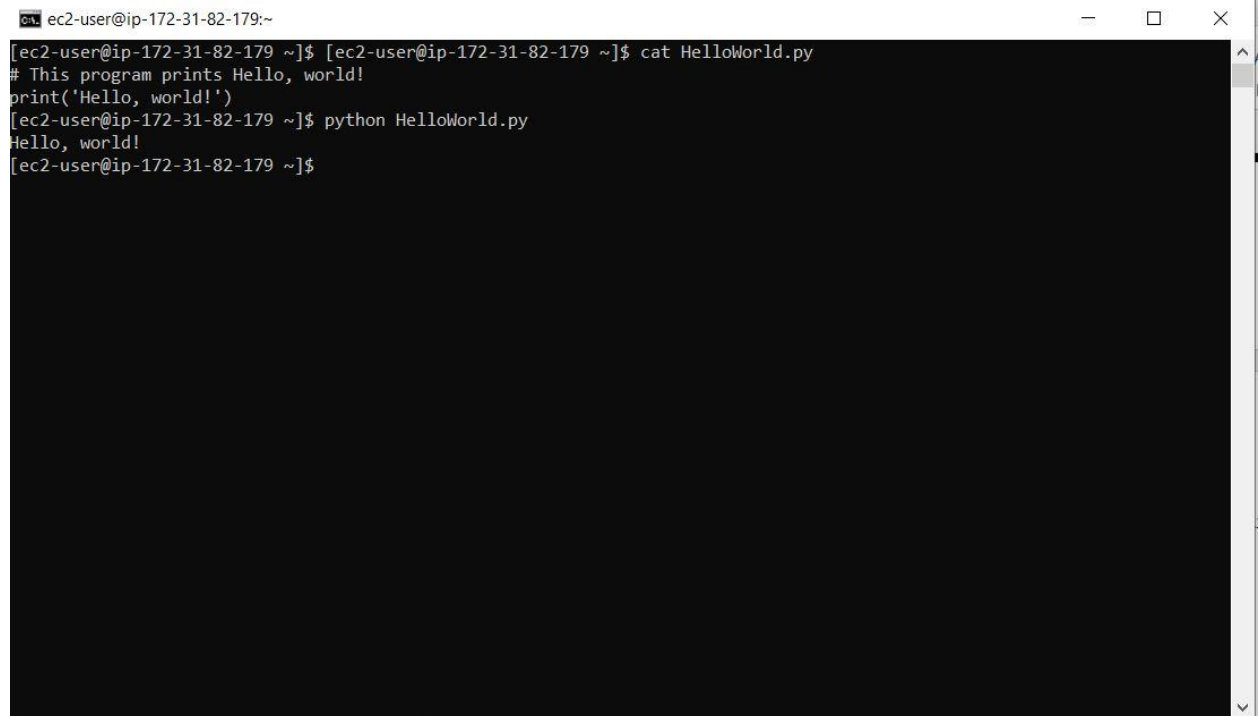
- c. Change the owner to you, disable inheritance and delete all permissions. Then grant yourself "Full control" and save the permissions. Now SSH won't complain about file permission to open anymore.



Success!



8. Create and run a python program: I made a simple hello world program!



```
ec2-user@ip-172-31-82-179:~  
[ec2-user@ip-172-31-82-179 ~]$ cat HelloWorld.py  
# This program prints Hello, world!  
print('Hello, world!')  
[ec2-user@ip-172-31-82-179 ~]$ python HelloWorld.py  
Hello, world!  
[ec2-user@ip-172-31-82-179 ~]$
```