Lab 1

Setting Up and Testing Amazon Web Services (AWS) Virtual Private Cloud (VPC)

CSCI 4417/5417-001

East Tennessee State University

Department of Computing

Fall 2015

Instructor: Jack Ramsey

Virtual Private Cloud Setup and Testing

Purpose

To install, configure, and test your AWS Virtual Private Cloud

Required

- Instructions

Conventions:

- Amazon Web Services = AWS
- Virtual Private Cloud = VPC
- Each component we configure will be named using your last name and descriptors specific for the component. For example, my VPC will be named RamseyVPC; its Internet gateway will be RamseyIGW, and so on.
- Your VPC IP Classless Inter-Domain Routing (CIDR) block will be 173.1.0.0/16.
- In this lab, *lastname* will appear where you are required to use your last name.

Set Up VPC

1. Log in to the AWS console
2. Configure shortcuts (EC2 & VPC)
3. Click Edit

Figure 1: Configuring Shortcuts (Step 3)

4. Click and drag EC2 to the shortcut bar

Figure 2: Adding the EC2 shortcut (Step 4)

Find the VPC shortcut and drag it to the shortcut bar

Figure 3: Adding the VPC shortcut (Step 4)

Click on Edit again to close the window

5. Make sure proper region is selected (upper right corner of the display)

Figure 4: Make sure we're in the proper AWS region

6. We'll go over this in lecture, but a little background may be in order: Amazon divides AWS into number of regional globally. Each region has at least two Availability Zones. An availability zone is a data center. AZs are located at geographically separate places within a region and are interconnected by high-speed networking. So data within a region can be replicated across AZs, providing fault tolerance in the event that one AZ goes down.

Figure 5: AWS's global infrastructure

7. Navigate to VPC by clicking on the shortcut
8. Click Your VPCs on the left menu
9. Click Create VPC
10. Name tag should be *lastname*VPC
11. CIDR block – 173.1.0.0/16.

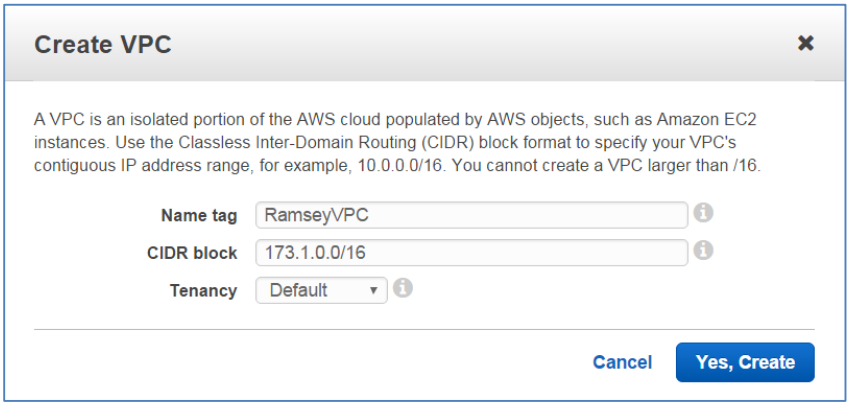


Figure 6: Creating a VPC (Steps 9 & 10)

12. Leave Tenency as Default – what that means is that you can make Amazon host all of the services on a single physical machine. For a fee. 'Default' is free
13. Click Actions
14. Click 'Edit DNS Resolution'. DNS Resolution should be set to 'Yes'

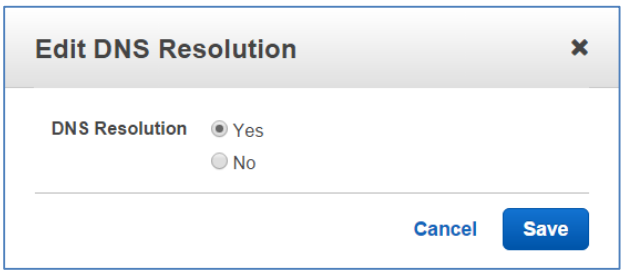


Figure 7: Confirm that the VPC will perform DNS resolution (it should be set to 'Yes'

15. Click Actions again
16. Select 'Edit DNS Hostnames'
17. Select 'Yes' and Save

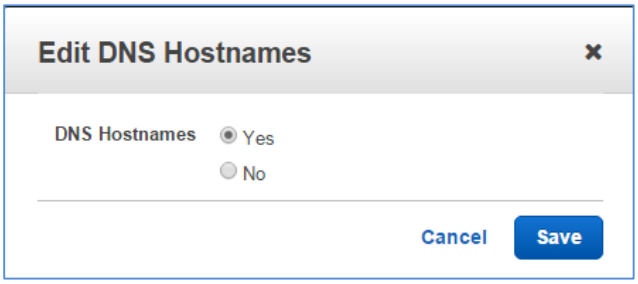


Figure 8: Enable DNS Hostnames (Step 16)

18. Click Subnets on the left menu and click on 'Create Subnet.'
19. Name – *lastname*PublicSubnet
20. Select appropriate VPC (*lastname*VPC)
21. Select us-east-1e availability zone
22. Enter CIDR block – 173.1.1.0/24 ('x' is your assigned number). How many potential addresses will be available (ignoring the fact that a few are reserved for DHCP, gateway, etc.)? Hint: There are 32 bits in an IPv4 address and we're using 24 of them for the netmask. So the potential addresses in this CIDR block go from 173.1.1.0 - 173.1.1.-what?

23. Click Yes, Create

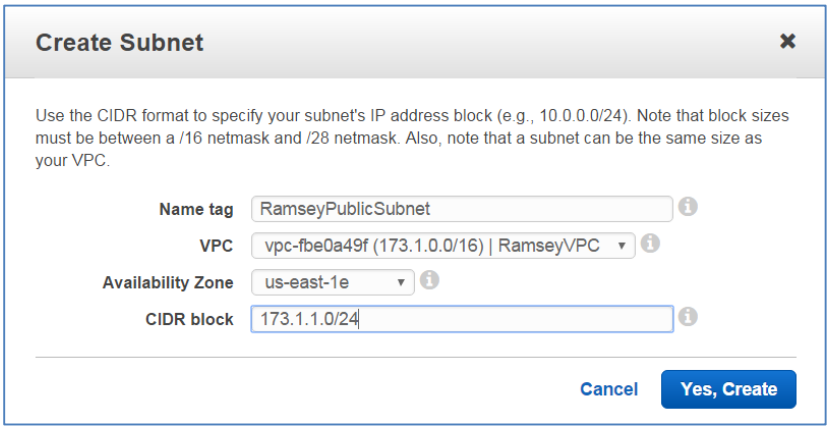


Figure 9: Subnet settings - Windows subnet (Steps 19-23)

24. Click Subnet Actions
25. Select Modify Auto-Assign Public IP
26. Check Enable auto-assign Public IP and Save

Figure 10: Enable auto-assignment of Public IP (Step 26)

27. Go to Internet Gateways on the left menu
28. Click on Create Internet Gateway
29. Name tag – *lastname*IGW

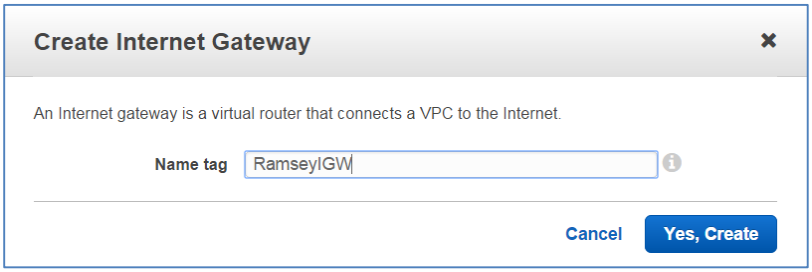


Figure 11: Naming an Internet Gateway (Step 28)

30. Select your IGW
31. Click on Attach to VPC
32. Select your VPC

33. Click on Yes, Attach
34. Go to Route Tables on the left menu
35. There will be two VPCs showing. The one you want to select is the one that has no subnets (173.1.0.0/16).
36. Click on Routes tab
37. Click on Edit
38. Click on 'Add another route'
39. Enter 0.0.0.0/0 (literally, "anywhere", a.k.a. '*Tha Innernet*')
40. Click in target field window
41. Select your IGW (when you click on the text field under 'Target,' the available Internet Gateway will appear. Each VPC can only have one IGW.
42. Click Save

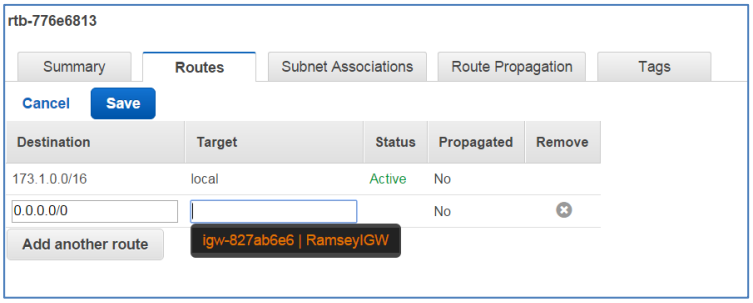


Figure 12: Associating IGW with a Route Table (Steps 35-41)

43. Click Subnet Associations tab
44. Click Edit
45. Select your new subnet
46. Click save

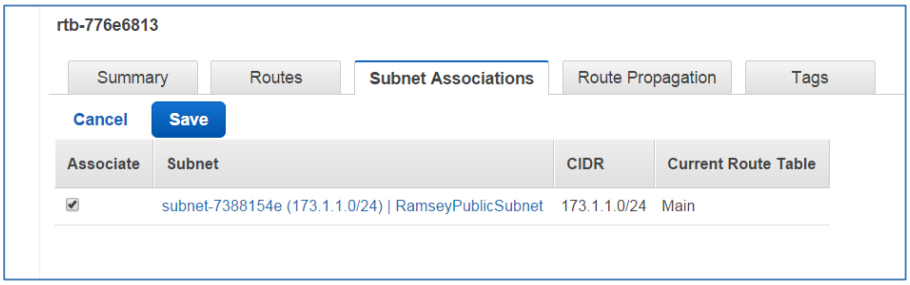


Figure 13: Associating subnets with a Route Table (Steps 42-45)

Congratulations! You have successfully set up your network environment. We hope. Let's test it, shall we?

Launch Ubuntu Instance and Connect

1. Navigate to the EC2 Dashboard by clicking on the shortcut you created earlier. (Hint: If you right-click the EC2 shortcut and select 'Open link in new tab,' you will then have your VPC dashboard in one tab and the EC2 dashboard in another, allowing you to toggle back and forth as needed)
2. Click on "Instances" in the left menu
3. Click the "Launch Instance" button to begin the process
4. Step 1 – Choose an Amazon Machine Image (AMI). We'll be launching an Ubuntu instance to test our VPC. Click the "Select" button to the right of "Ubuntu Server 14.04 LTS (HVM), SSD Volume Type – ami-fce3c696"

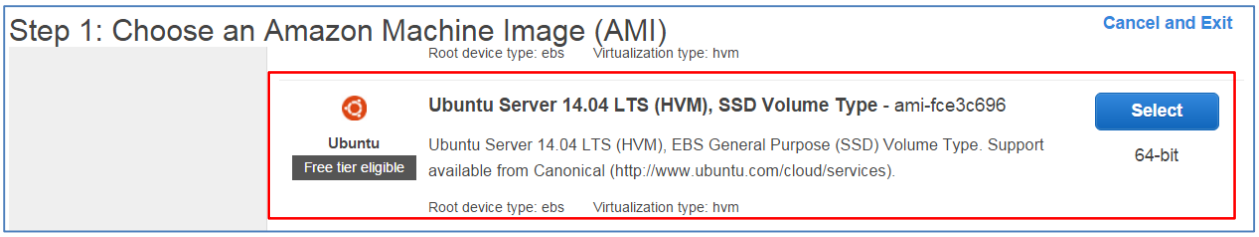


Figure 14: Look for Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-fce3c696. It's the fourth one on the list

5. Step 2: Choose an Instance Type – "t2 micro" should already be selected. Select it, if not. Click the "Next: Configure Instance Details" button
6. Step 3: Configure Instance Details – Several things need to be done, here:
   a. Network – Choose your VPC from the dropdown list

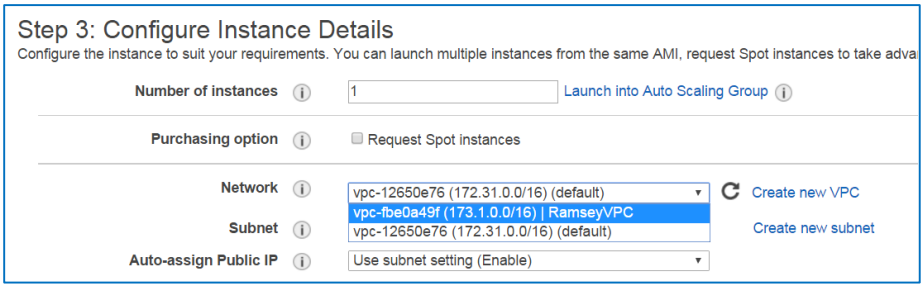


Figure 15: Selecting a VPC (Step 6)

   b. Subnet – The subnet you just made should be displayed. Later, when we have multiple subnets, you have to specify which one new instances will be connected to

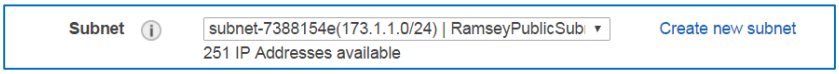


Figure 16: Choosing a subnet (Step 6.b)

   c. Make sure Auto-assign IP says either "Enable" or "Use subnet setting (Enable)"
   d. In the Primary IP window at the bottom of the page, enter 173.1.1.10 to create a static IP address for the instance. Interestingly, the operating system will still think it is

obtaining its IP address dynamically, but AWS is handling things behind the scenes

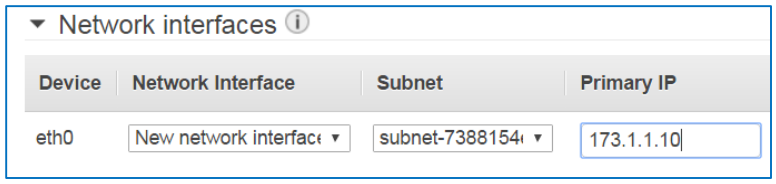


Figure 17: Assigning static IP address (Step 6.d)

e. Click the "Next: Add Storage" button

7. Step 4: Add Storage – 8 GiB is plenty. Click the "Next: Tag instance" button

8. Name your instance *lastname*UbuntuTest. Click the "Next: Configure Security Group" button

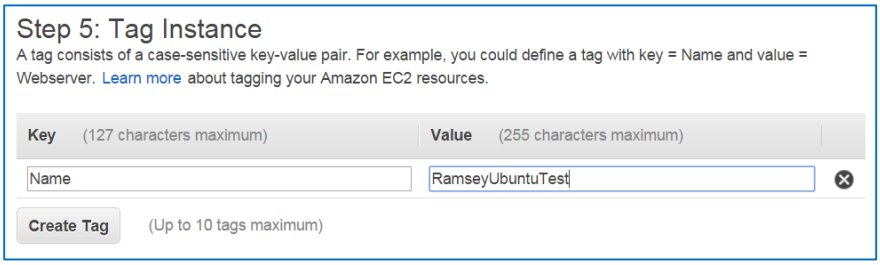


Figure 18: Naming the instance (Step 8)

9. In the Security group name: field, enter *lastname*SG. Add a description. A security group is Amazon's virtual version of a firewall. You can explicitly specify both IP address ranges and port numbers to allow or block

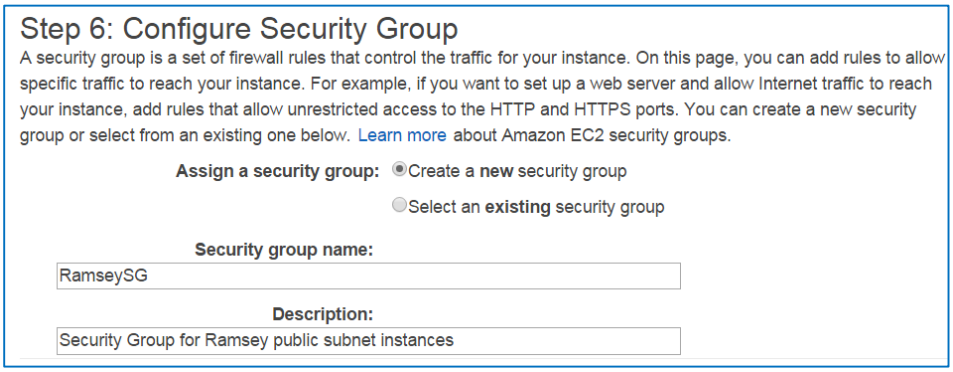


Figure 19: Creating a new Security Group

10. Since we'll use this security group for both Windows and Ubuntu instances, we need to modify the security group rules. Click on 'Add Rule'

11. Click the drop-down list under 'Type'

12. Scroll down to RDP

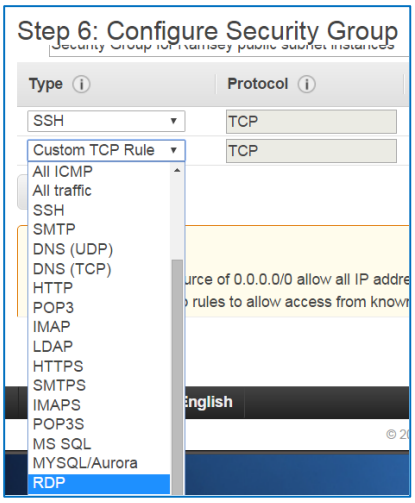


Figure 20: Opening the Remote Desktop Protocol port

13. Under 'Source,' click the drop-down list and select 'Anywhere.' That should do it for now. If we need to modify the security group later, we can. We may, for example, decide to install an Apache web server, which would mean opening port 80 so we could connect to it

14. Click the "Review and Launch" button. Ignore the dire warning at the top of the display. Obviously, if this were a production machine, we would want our security group to be much more restrictive. For example, we could have restricted access to be only from campus by choosing a custom IP address (instead of 'Anywhere') and enter '151.141.0.0/16'

15. Click the "Launch" button on the following display.

16. In the dialog window that appears, the 'Create a new key pair' option should be selected. Enter '4417key' into the 'Key pair name' field. Click the acknowledgement checkbox and click on 'Download Key Pair.' Save the key, 4417key.pem, to your external drive. Don't save it on the Z:\ drive. Make sure you store it in a location you will remember! The key cannot be recovered if you lose it! Then click the "Launch Instances" button.

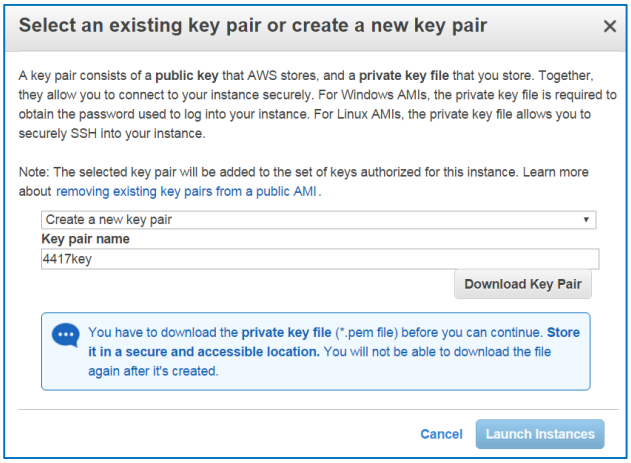


Figure 21: Assigning key pair to the instance and launching (Step 11)

17. Click the “View Instances” button. This will return you to the AWS EC2 Dashboard.
18. While the instance is launching, let’s convert the “4417keypair.pem” file to a format that PuTTY can use to secure shell (SSH) in to our new instance. Launch PuTTYgen.exe.
19. Click on the “Load” button.
20. Select “4417keypair.pem” from the location you saved it (you’ll have to change the file type in the lower right corner of the window to “All Files (*.*) for it to display.
21. Click on ‘OK’ in the dialog that displays, then the “Save private key” button to save your .ppk file. Name the file “4417key.ppk” (ppk = “Putty Private Key”, while pem = “Privacy Enhanced Mail”) and save it to the same location as “4417key.pem.”

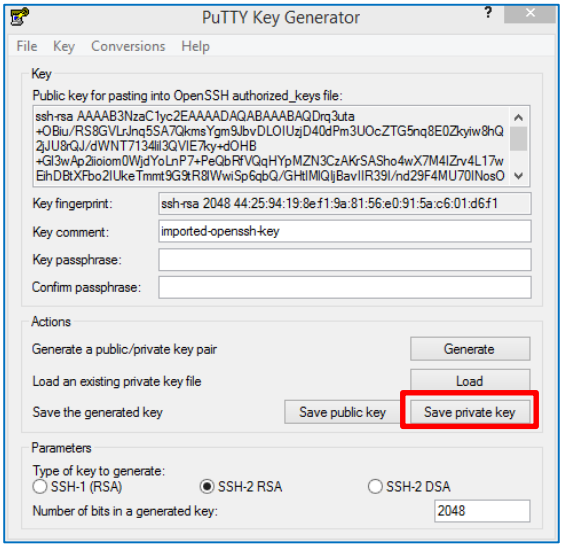
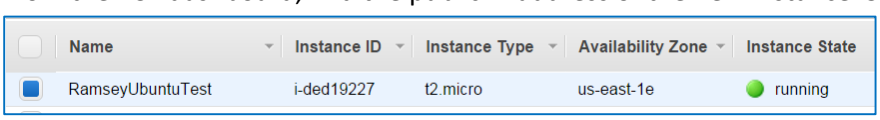


Figure 22: Saving the private key generated from the .pem file (Steps 18-21)

22. From the EC2 dashboard, find the public IP address of the new instance. Select the instance,



Name	Instance ID	Instance Type	Availability Zone	Instance State
RamseyUbuntuTest	i-ded19227	t2.micro	us-east-1e	running

Figure 23: Selecting a running instance

23. …and look at the right column of the lower pane for “Public IP.” Select and copy this value

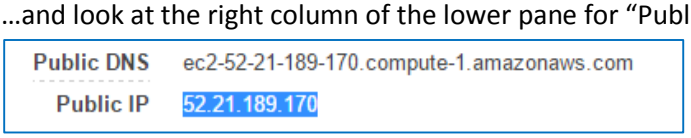


Figure 24: Test instance's public IP address (Step 23)

24. Launch PuTTY
25. First, to make things a little easier in the future, click on the ‘+’ sign next to ‘SSH’ in the left menu tree. Select ‘Auth’
26. In the right pane, click on ‘Browse’ and navigate to the location that you stored your keys

27. Select '4417key.ppk'

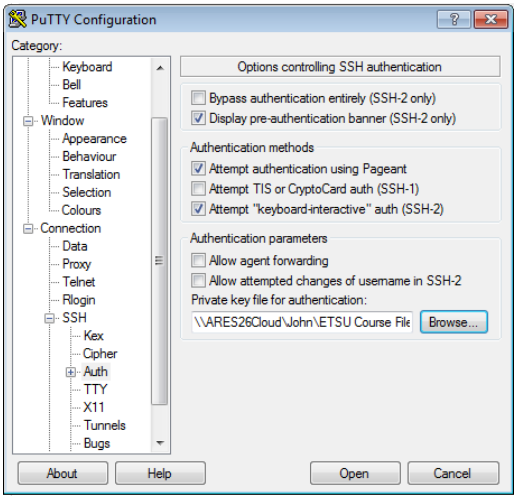


Figure 25: Saving a 'Key Session' (Step 26)

28. In the menu tree, select 'Session' (at the top of the tree)
29. In the 'Saved Session' field, enter '4417key'
30. Click 'Save'. Now, when you need to SSH into an instance, you can first load the 4417key session, by selecting it from the list and clicking the 'Load' button, which will have your key pre-selected
31. Paste the instance's public IP address in the "Host Name (or IP address)" field

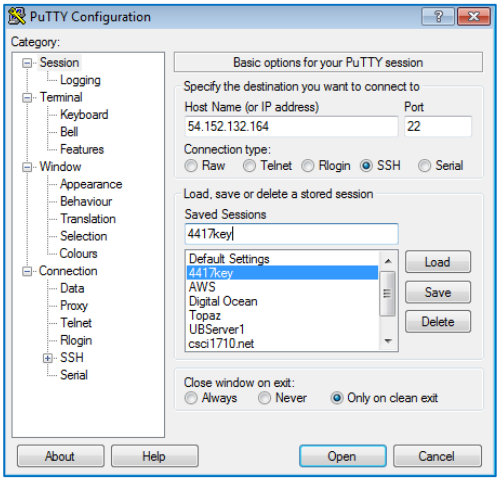


Figure 26: Using PuTTY to log in to an instance

32. Click the "Open" button
33. Click the "Yes" button on the security warning dialog
34. Login as user "ubuntu" (without the quotation marks)
35. You should log in to your new server and see the command prompt. If you want to verify connectivity, enter

```
ping -c4 yahoo.com
```

(the "-c4" switch limits the command to four pings, like Windows' **`ping`** does by default. With Windows, if you want, you can make it ping continuously using the /t switch). Of course, if we're able to connect to the server, we know that it is connected to the Internet, but it's nice to do something with it after all this work

36. Enter

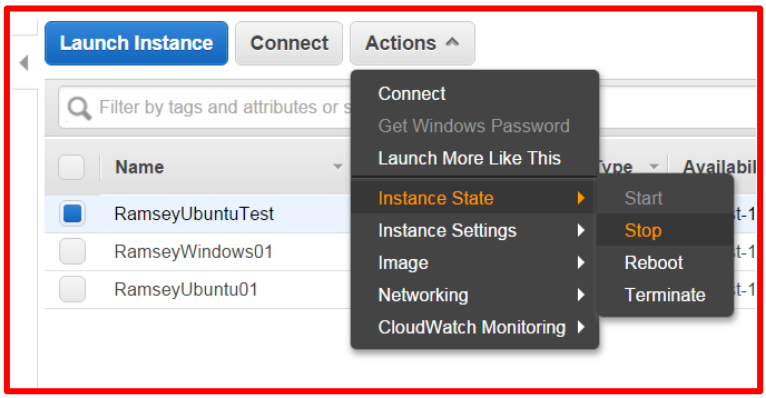
```
sudo shutdown now
```

to stop your server.

For your lab report submission, prepare a summary of today's activities. Be sure to note anything that you had trouble with. What did you think was important? If you were sitting next to a friend who had never used AWS, how would you talk them through what we just did (just hit the high points. Otherwise you'll simply be recreating this torturously long document). The report will be due by class time next Tuesday.

Feel free to play with the service – launch a (free tier) Windows Server 2012 instance if you wish. Try to figure out how to remote connect to it (you'll be using Remote Desktop Connection instead of PuTTY, but you have to get the password that's generated by using the "4417keypair.pem" file). None of this is required, but you'll benefit from AWS the more you use it. Amazon provides instructions – hint: right-click on the instance in the instance list and select "Get Windows Password"

ALWAYS MAKE SURE THAT ALL INSTANCES ARE STOPPED BEFORE LOGGING OUT OF AWS! BEFORE YOU LEAVE LAB, MAKE SURE THE EC2 DASHBOARD INDICATES THAT ALL INSTANCES' STATES ARE 'STOPPED'! SOMETIMES THE DASHBOARD IS SLOW TO UPDATE (SINCE WE STOPPED IT FROM THE CLI IN STEP 35. YOU CAN CLICK ON 'ACTIONS,' MOUSE DOWN TO 'INSTANCE STATE, MOUSE OVER TO 'STOP' AND CLICK.



If you do launch additional instances, and I encourage you to do so, please ensure you're using the "free tier" level and that you shut your instances down once you're done. Leave your new VPC and instances in place once you are done with the required steps (we'll use them in lab next week).