

**IaaS – Getting started with AWS EC2**

**Objective:** Get practical experience in using Amazon AWS EC2 system to create and configure Linux/Windows virtual instances as a web server.

**Tasks:**

1. Creating Windows Server VM

2. Creating Linux Ubuntu VM

3. Configure RDP and SSH connections to VMs

**Lab environment**:

* Home/university PC (Microsoft Windows / Linux / MacOS)
* RDP client (for Linux/MacOS)
* SSH or PuTTY (for Windows)

***Contents***

[Overview 3](#_heading=h.gjdgxs)

[Step 1: Logging in the AWS Management Console 3](#_heading=h.30j0zll)

[Step 2: Running Windows Server VM on Amazon EC2 6](#_heading=h.1fob9te)

[Launch a Windows Server Virtual Instance 6](#_heading=h.3znysh7)

[Browse the Web Server 13](#_heading=h.2et92p0)

[Step 3: Connecting To Your Windows Instance 15](#_heading=h.3dy6vkm)

[Step 4: Running Linux Server VM on Amazon EC2 20](#_heading=h.4d34og8)

[Launch a Linux Server Virtual Instance 20](#_heading=h.2s8eyo1)

[Browse the Web Server 26](#_heading=h.17dp8vu)

[Step 5: Connecting to Your EC2 Linux Instance 28](#_heading=h.3rdcrjn)

[Converting .pem private key to .ppk private key used by PuTTY 28](#_heading=h.26in1rg)

[Connecting to Linux VM using PuTTY 29](#_heading=h.lnxbz9)

[Step 6: Terminate instances after use!!! 31](#_heading=h.35nkun2)

[References 32](#_heading=h.1ksv4uv)

# Overview

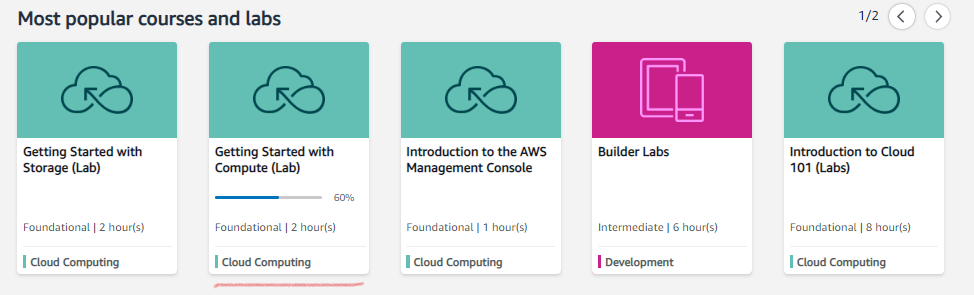
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.

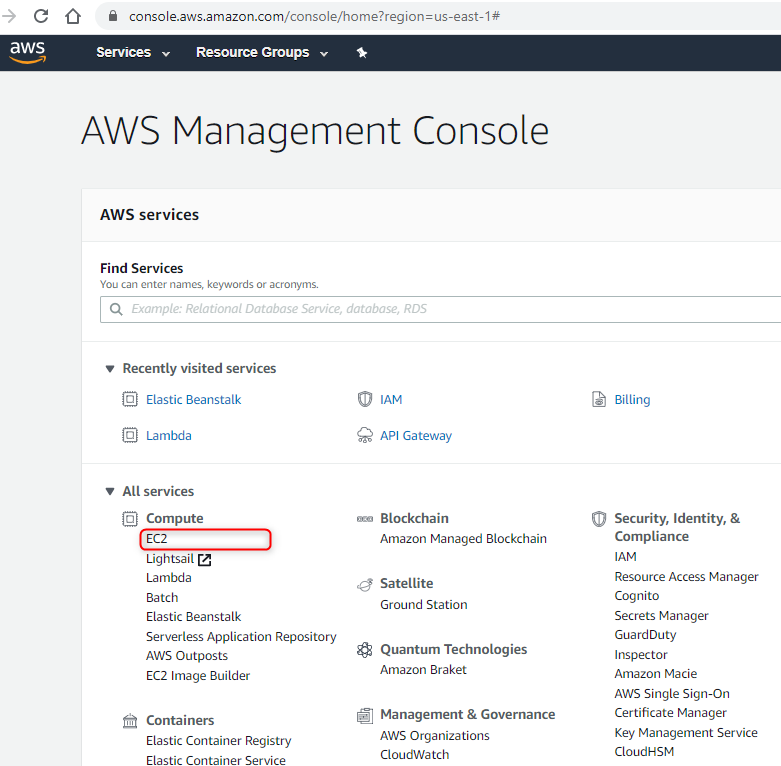
This lab will walk you through launching, configuring, and customizing an EC2 servers using the AWS Management Console.

# Step 1: Logging in the AWS Management Console

1. Create a free tier AWS account which will give you 12 free access to almost all AWS resources as instructed [here](https://docs.google.com/document/d/1r-F3RqO_6QaVPvktp_vs29G-WvoRPuLr/edit?usp=sharing&ouid=108921260211585580927&rtpof=true&sd=true) if you have not done so yet.
2. Sign into your AWS free tier account and open the Amazon EC2 console at <https://aws.amazon.com/console/>

**Alternatively**, if you have not completed your AWS registration yet, but have already registered with AWS Educate (which does not require any credit/debit card) you can complete this lab at AWS Educate. For this purpose, sign in to AWS Educate (<https://aws.amazon.com/education/awseducate/>), select ‘Getting Started with Compute (Lab)’ in the list of most popular courses and labs and follow the instruction. This activity is pretty similar to the lab tasks discussed above. Though, you will only be given 3 hour access to AWS console after you start the lab. .



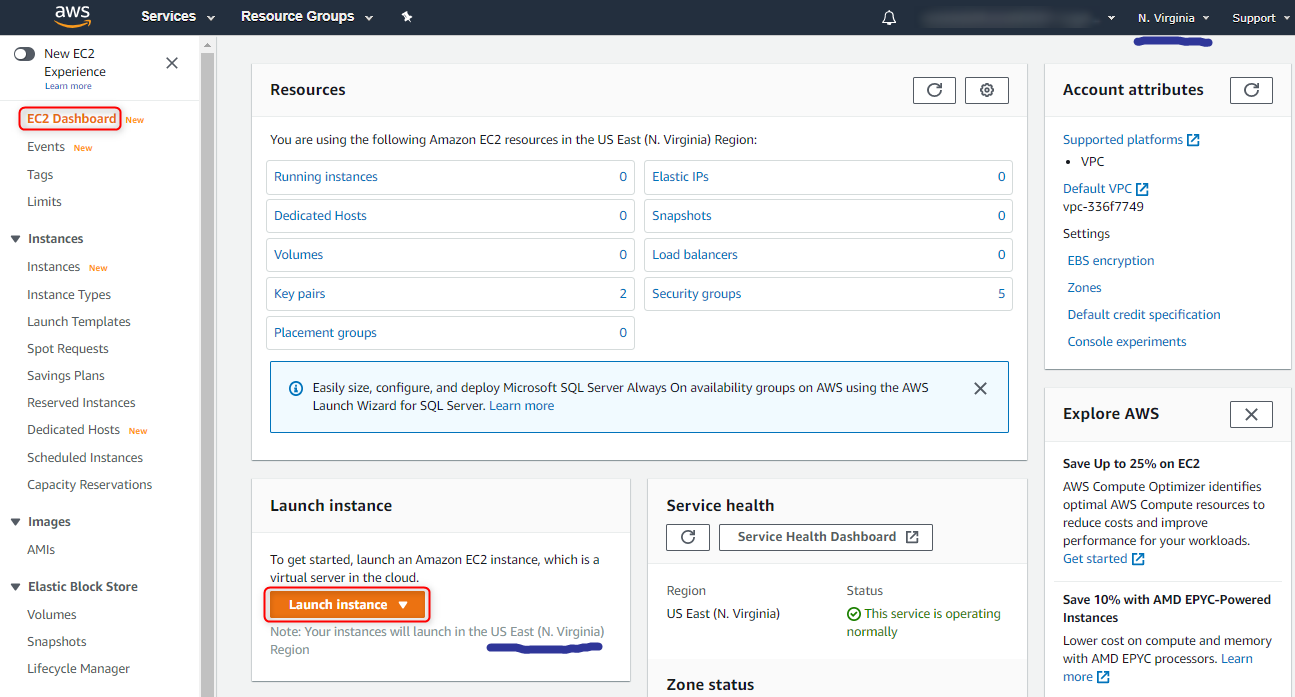


# Step 2: Running Windows Server VM on Amazon EC2

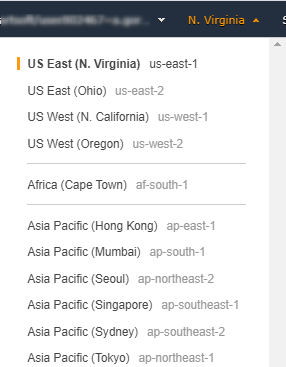
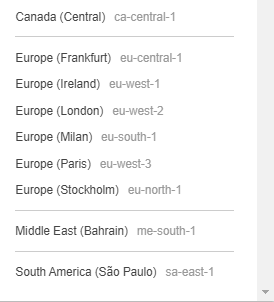
## Launch a Windows Server Virtual Instance

In this example we will launch a Windows Server 2012 R2 instance with the IIS web server installed upon boot.

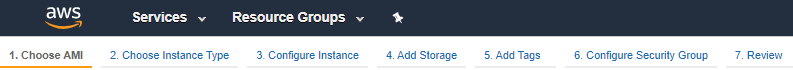
1. Once you opened the Amazon EC2 console in the AWS Management Console, click on **Launch Instance**



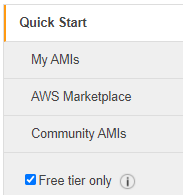
**Notice** the Region your instance will launch in. You can change the region in the top right menu. There are more than 20 regions AWS available across US, Africa, Asia, Canada, Europe, and Middle East.

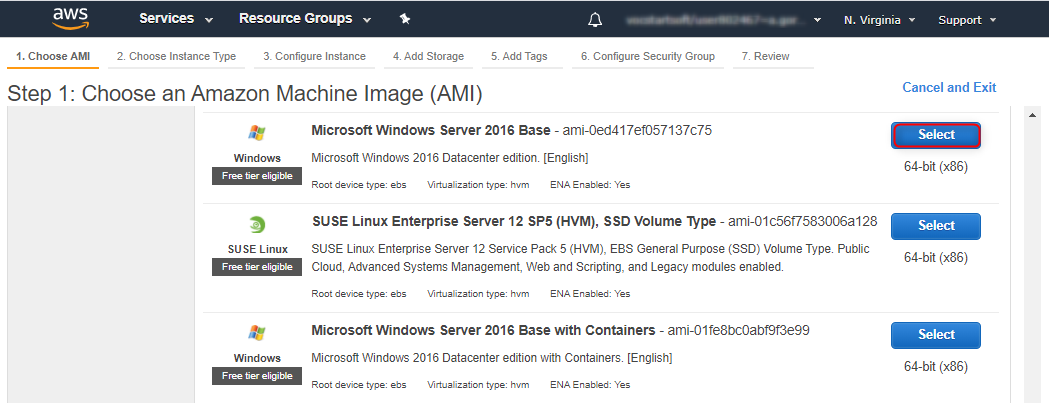
On the top of your screen you can notice the general workflow of creating a VM instance. It includes Choosing Amazon Machine Image (AMI), Choosing Instance Type, Configuring the Instance, Adding Storage, Adding Tags, Configuring Security Groups and Reviewing:



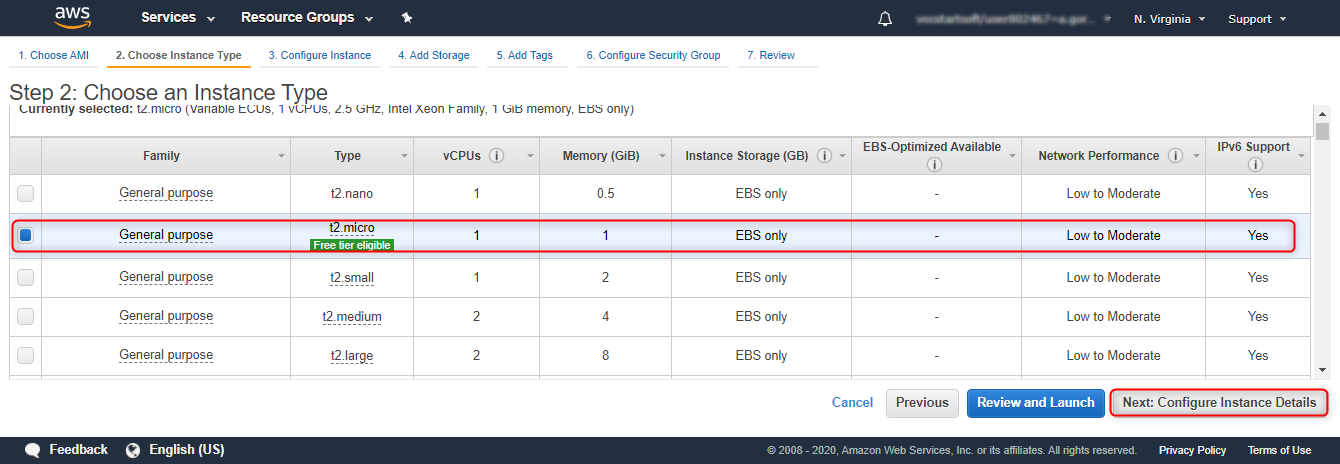
1. On the left panel pick out “**Free tier only**” to display only those VM images which are eligible for the free tier usage.



1. Scroll down and click **Select** on the Windows Server 2016 R2 Base AMI.



1. In the **Choose Instance Type** tab, select the **t2.micro** instance size and click **Next: Configure Instance Details**



1. On the **Configure Instance Details** page, expand the **Advanced Details** section, copy/paste (when copy/paste from Google Drive some special characters, like ‘<’, ‘>’, etc. could be replaced with their codes; thus, it is recommended to download the labsheet on your PC and open in MS Word from where you can copy/paste the code) the script into the **User Data** field:



<powershell>

Install-WindowsFeature Web-Server -IncludeManagementTools -IncludeAllSubFeature

remove-item -recurse c:\inetpub\wwwroot\\*

(New-Object System.Net.WebClient).DownloadFile("https://github.com/anatoliy-gorbenko/CloudComputing/blob/main/ec2-windows.zip?raw=true", "c:\inetpub\wwwroot\ec2-windows.zip")

$shell = new-object -com shell.application

$zip = $shell.NameSpace("c:\inetpub\wwwroot\ec2-windows.zip")

foreach($item in $zip.items())

{

$shell.Namespace("c:\inetpub\wwwroot\").copyhere($item)

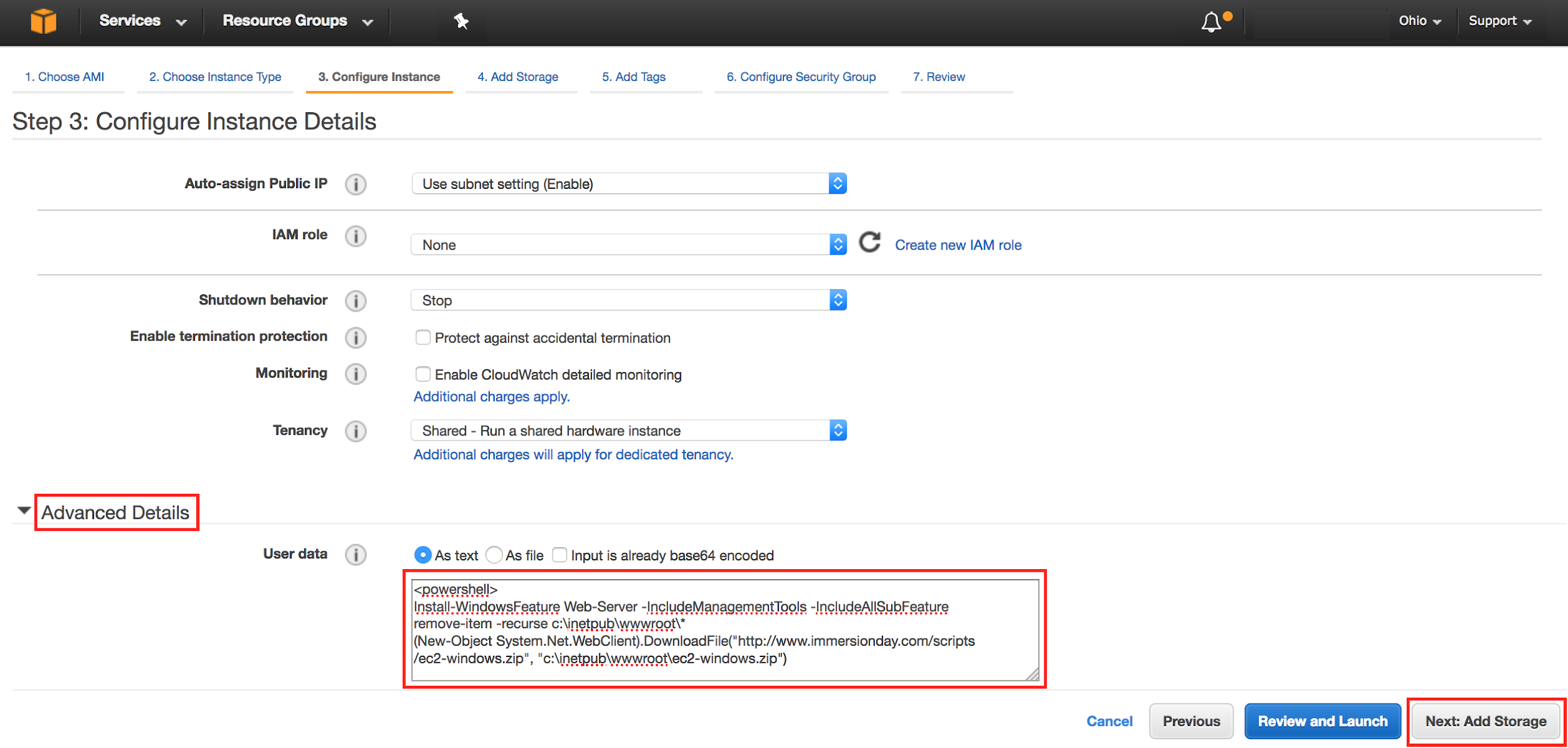
}

</powershell>

This PowerShell script will install/start IIS and deploy a simple web page by downloading and unzipping ec2-windows.zip file with the sample .aspx and .css files into a wwwroot folder of the IIS web server.

You can also do the same manually by configuring your Windows Server after you launch the VM and get access to it.

For further information on User Data please refer to the documentation at <http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-instance-metadata.html>.



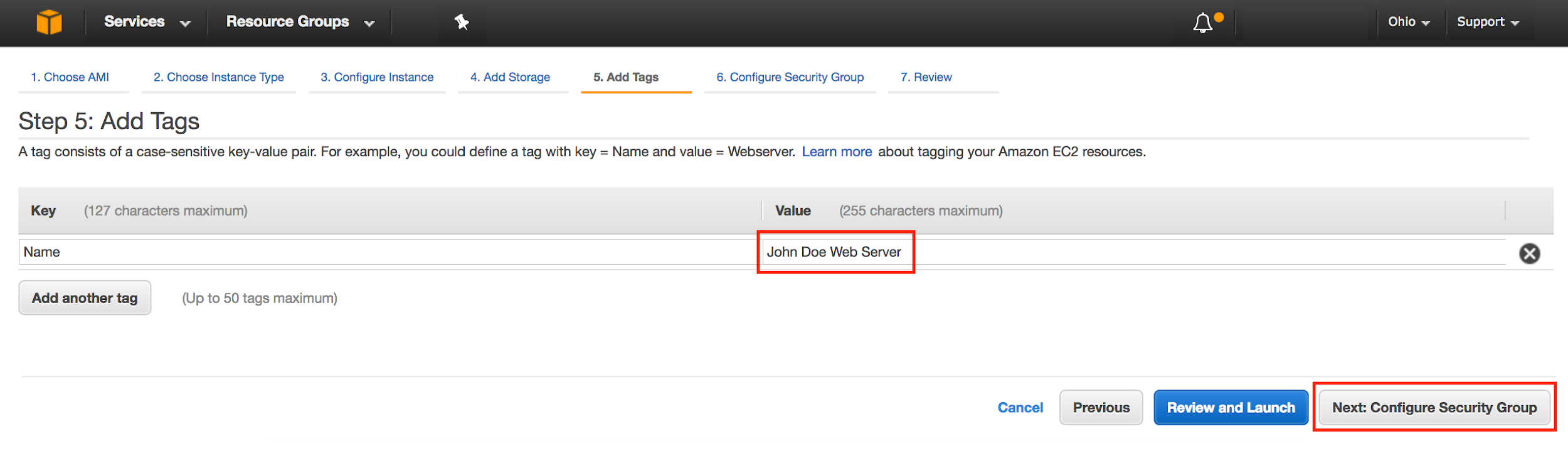
1. Click **Next: Add Storage**:

On the **Step 4: Add Storage** screen, Click **Next: Add Tags** to accept the default Storage Device Configuration and move to the **Step 5: Add Tags screen**.

Next, choose a “friendly name” for your instance. This name, more correctly known as a tag, will appear in the console once the instance launches. It makes it easy to keep track of running machines in a complex environment.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with **Key** = “**Name**” and **Value** = “**[Your Name] Web Server**”.

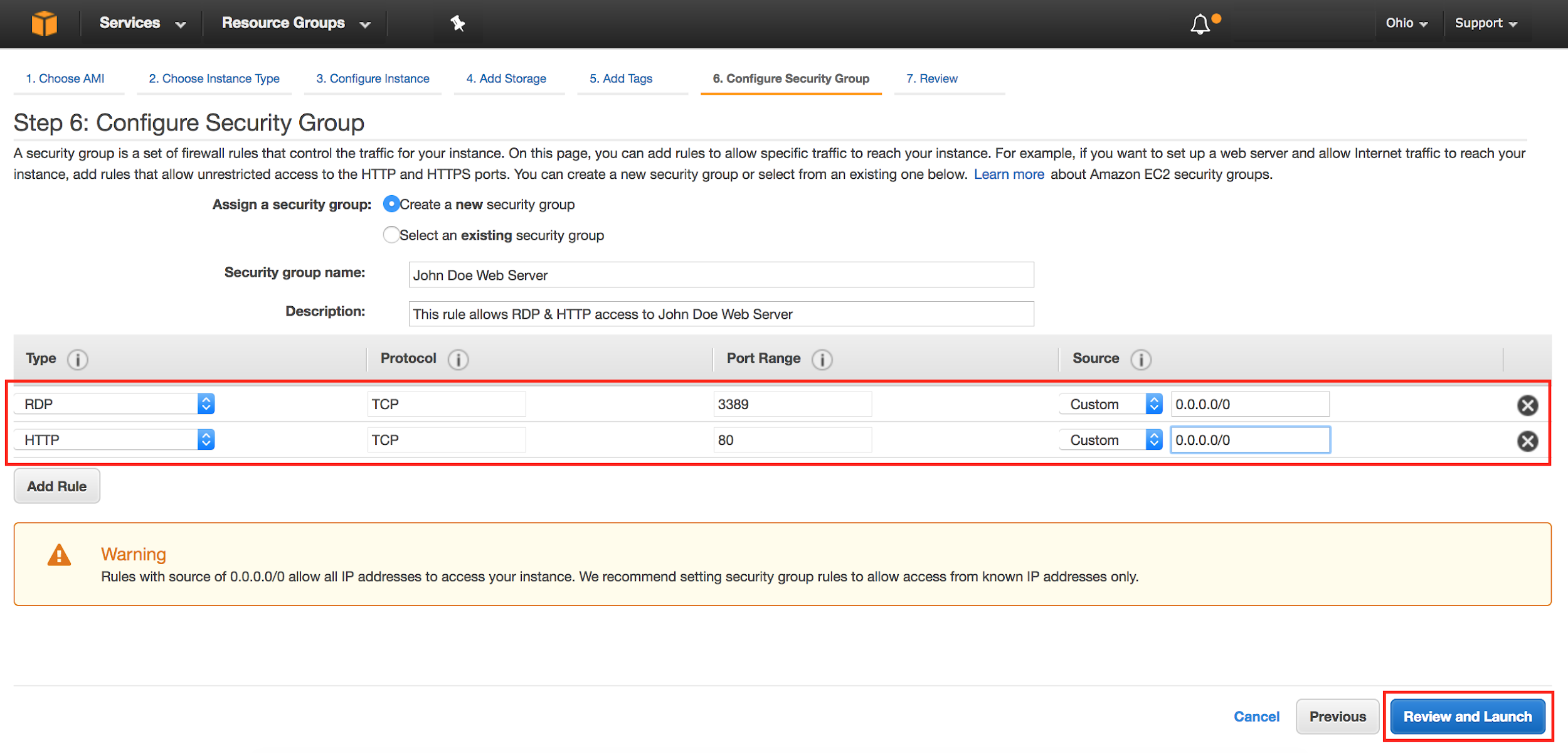
Then click **Next: Configure Security Group**.

  
For further information on Tags please refer to the documentation at - <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html>.

1. You will be prompted to create a new security group, which will be your firewall rules. On the assumption that we are building out a Web server, name this security group according to this format **“[Your Name] Web Server**”.

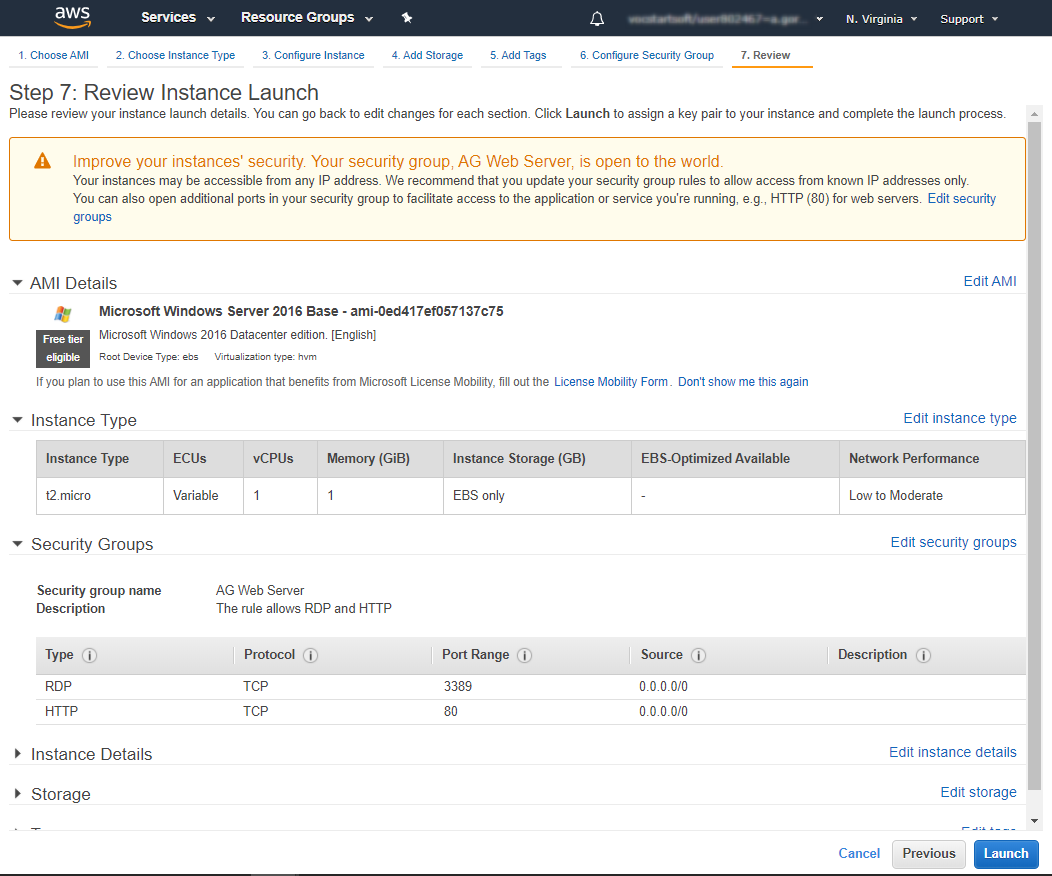
**Add Rule** to open port 80 (HTTP) in addition to 3389 (RDP).

Click the **Review and Launch** button after configuring the security group.

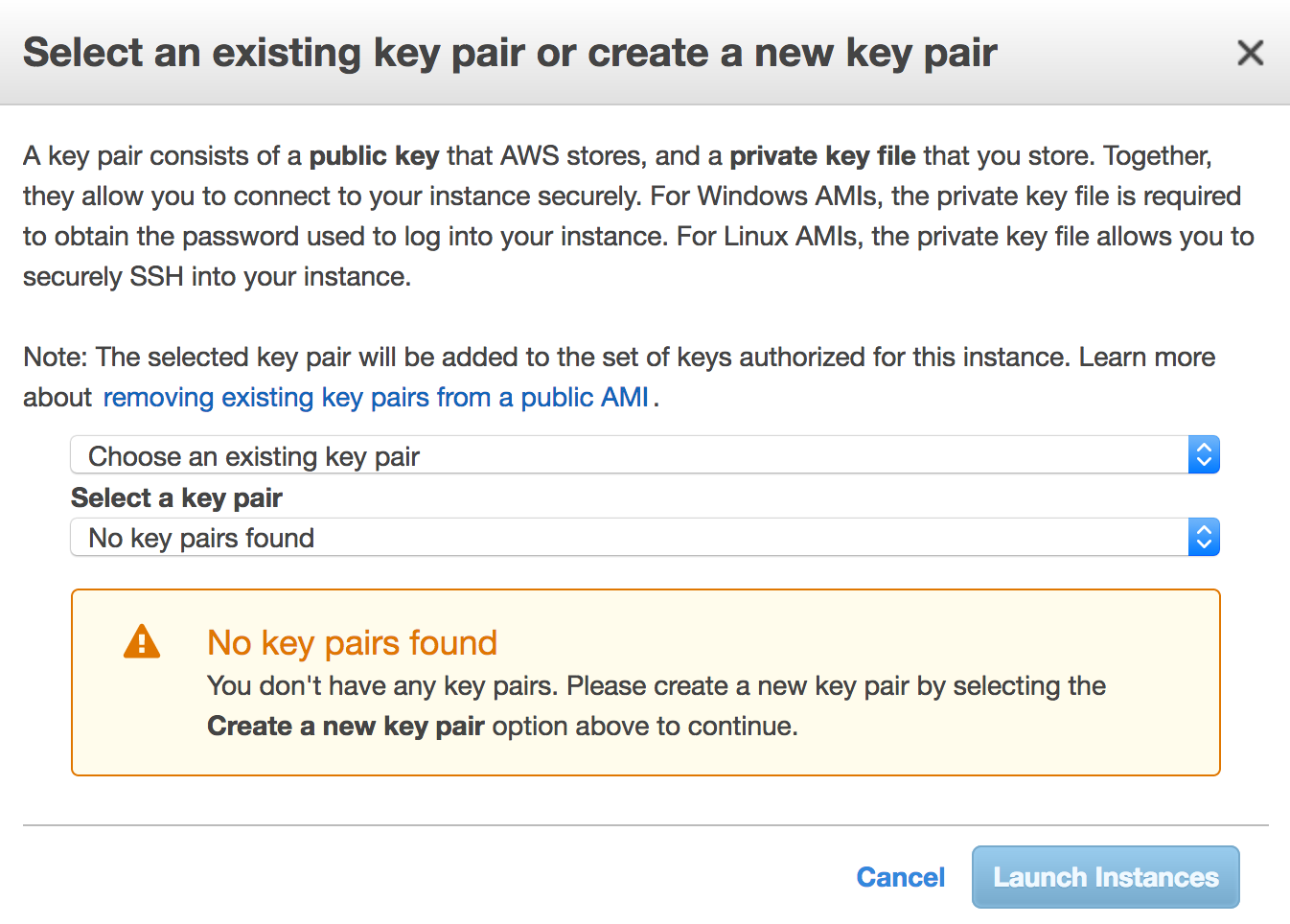


1. Review your choices, and then click **Launch**.

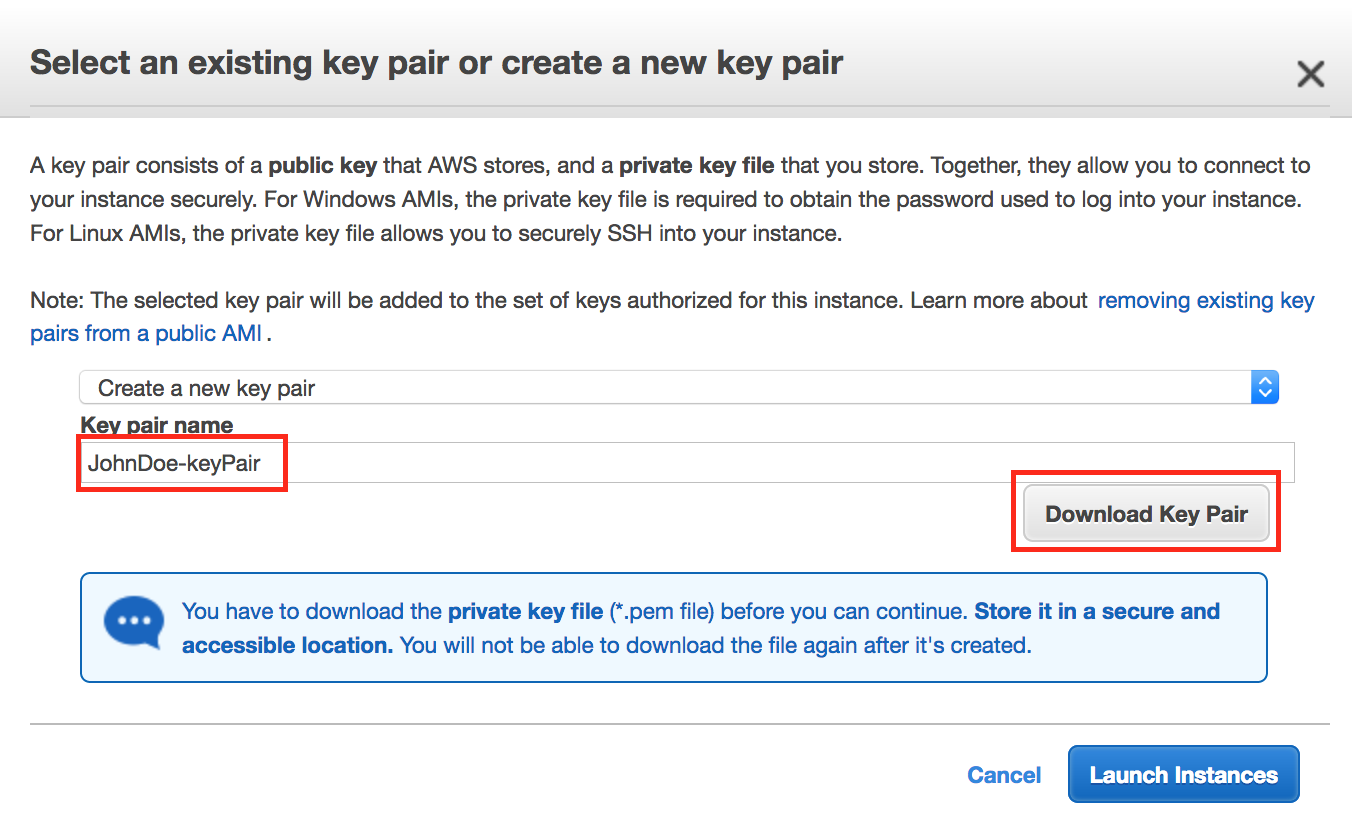
***Note*** *the warning boxe(s) at the top of the page, this is to warn you about possible configuration issues. In this lab we are creating a Windows server that has RDP and HTTP access that is “open to the world” this is something that you wouldn’t normally do.*



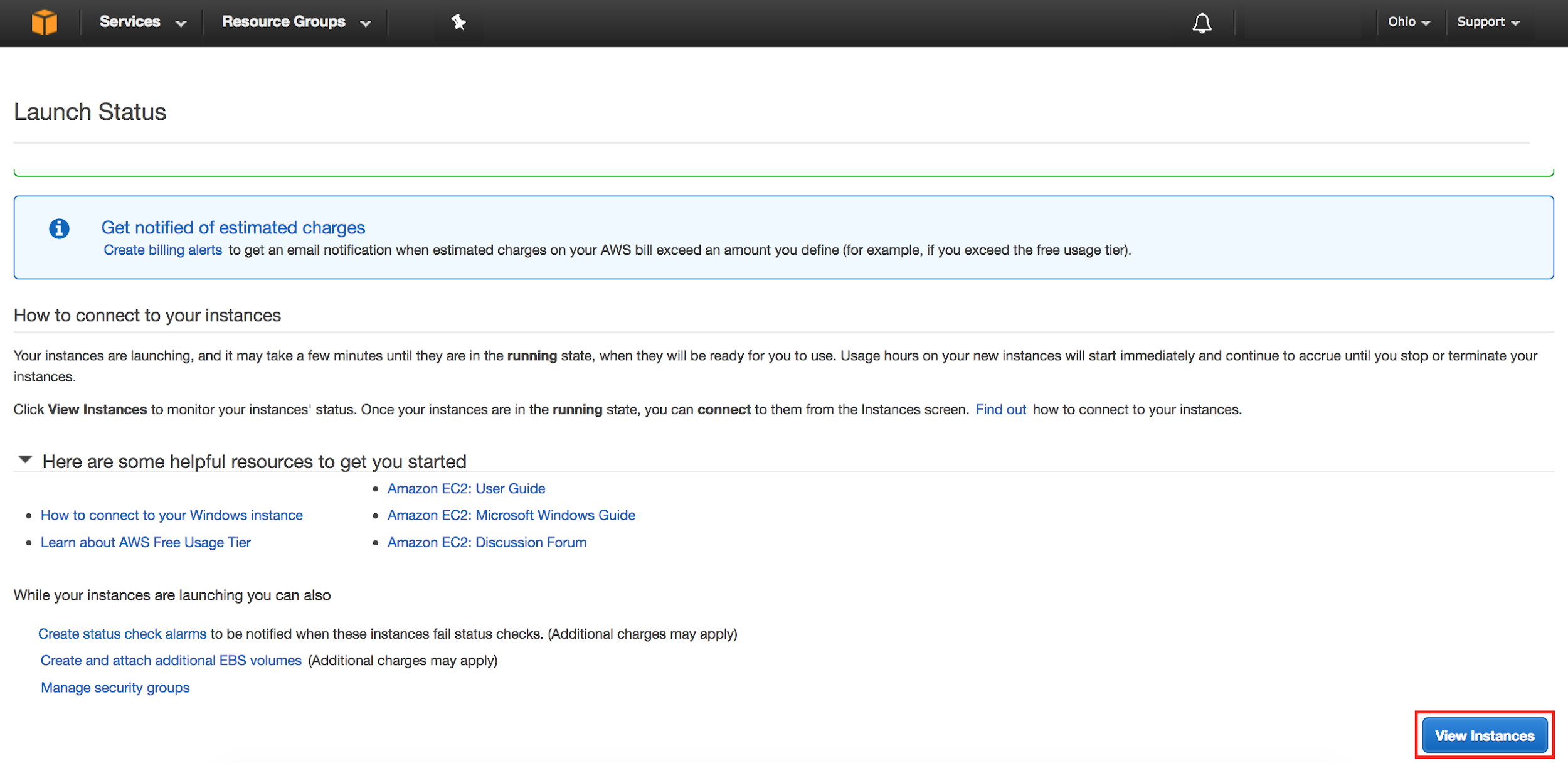
1. Now you need to create a public/private keypair. When this instance launches, you will connect to it via Remote Desktop using the credentials for “administrator”. For Windows instances, EC2 automatically generates a password and encrypts with your public key. To decrypt the encrypted password, you will use your private key. Let’s create a new public/private keypair.



1. Enter a name for the key pair (e.g. using the following format: [YourName]-KeyPair) and click **Download Key Pair**.



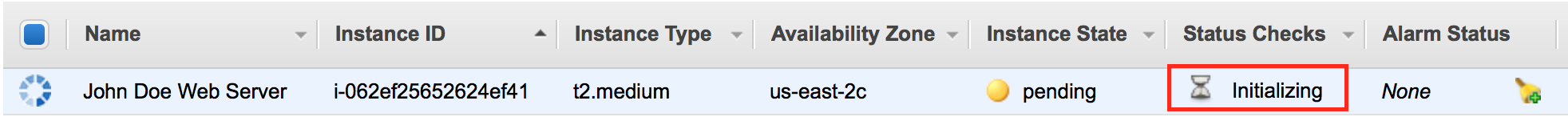
1. Your browser will download the private portion of the key pair to your PC. It will have a name like *JohnDoe-keyPair.pem*. **Note** the location of this file because you will need it later to decrypt the administrator password. It might be a good idea to save the file on a memory stick or send it to your email/save on your Google Drive so you can use the same key pair later in the module.
2. Now click the **Launch Instances** button to launch your Windows web server.
3. The next screen will confirm that your instance is now launching. Click the **View Instances** button. Once your instance has launched, you will see the “**[Your Name] Web Server**” instance as well as the Availability Zone the instance is in and its publicly routable DNS name.



## Browse the Web Server

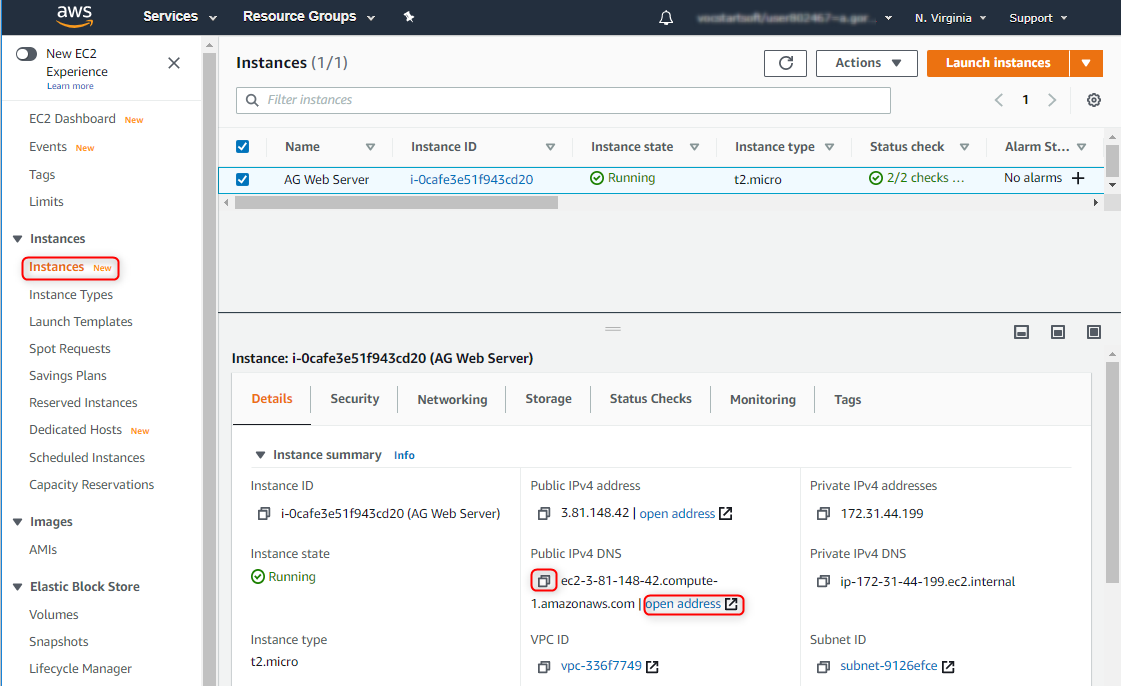
Now you will browse to the Web Server site that was installed on the Instance using the PowerShell script defined in the **User Data** section during creation of the instance.

1. Wait for the instance to pass the Status Checks. For Windows instances, this could take up to 20 minutes.



When complete, you will see the Status Checks have passed.

From the window below you can also notice your VM’s IP address and Public DNS name.



1. Click on **Open address** or open a new browser tab and browse the Web Server by entering the EC2 instance’s Public DNS name into the browser (make sure you use ’http://’ prefix, not ‘https://’). You should see a page that looks similar to this (if not, wait for some time and reload the page):



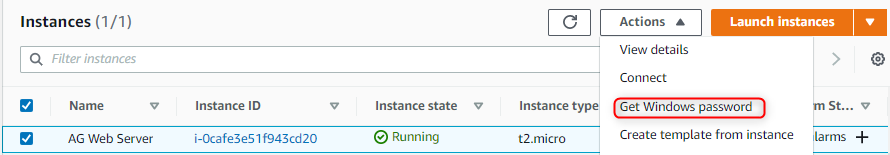
Great Job: You have built your first Windows web server!

# Step 3: Connecting To Your Windows Instance

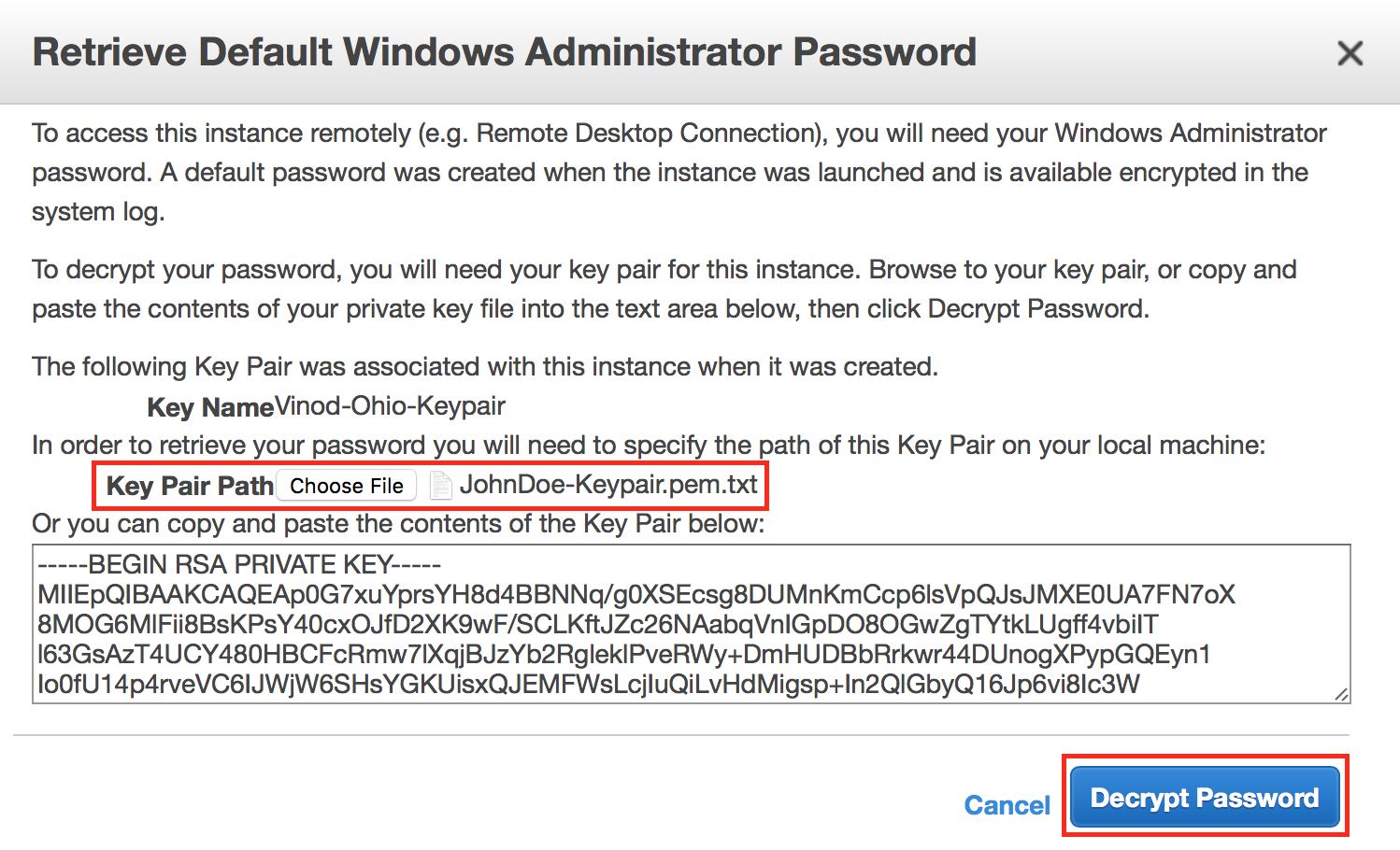
An RDP client needs to be used to connect to the Windows desktop.

On a Windows PC use the bundled *Remote Desktop* application. For Mac users, if you don’t have a RDP client already installed, download and install the [*Microsoft Remote Desktop*](https://itunes.apple.com/us/app/microsoft-remote-desktop/id715768417?mt=12&ls=1). If your home PC’s is Linux you can use one of the following RDP clients: [*Remmina*](https://kamarada.github.io/en/2020/04/20/remote-desktop-connection-to-windows-from-linux-using-rdp-clients/#remmina), [*FreeRDP*](https://kamarada.github.io/en/2020/04/20/remote-desktop-connection-to-windows-from-linux-using-rdp-clients/#freerdp), [*rdesktop*](https://kamarada.github.io/en/2020/04/20/remote-desktop-connection-to-windows-from-linux-using-rdp-clients/#rdesktop) or [*Vinagre*](https://kamarada.github.io/en/2020/04/20/remote-desktop-connection-to-windows-from-linux-using-rdp-clients/#vinagre).

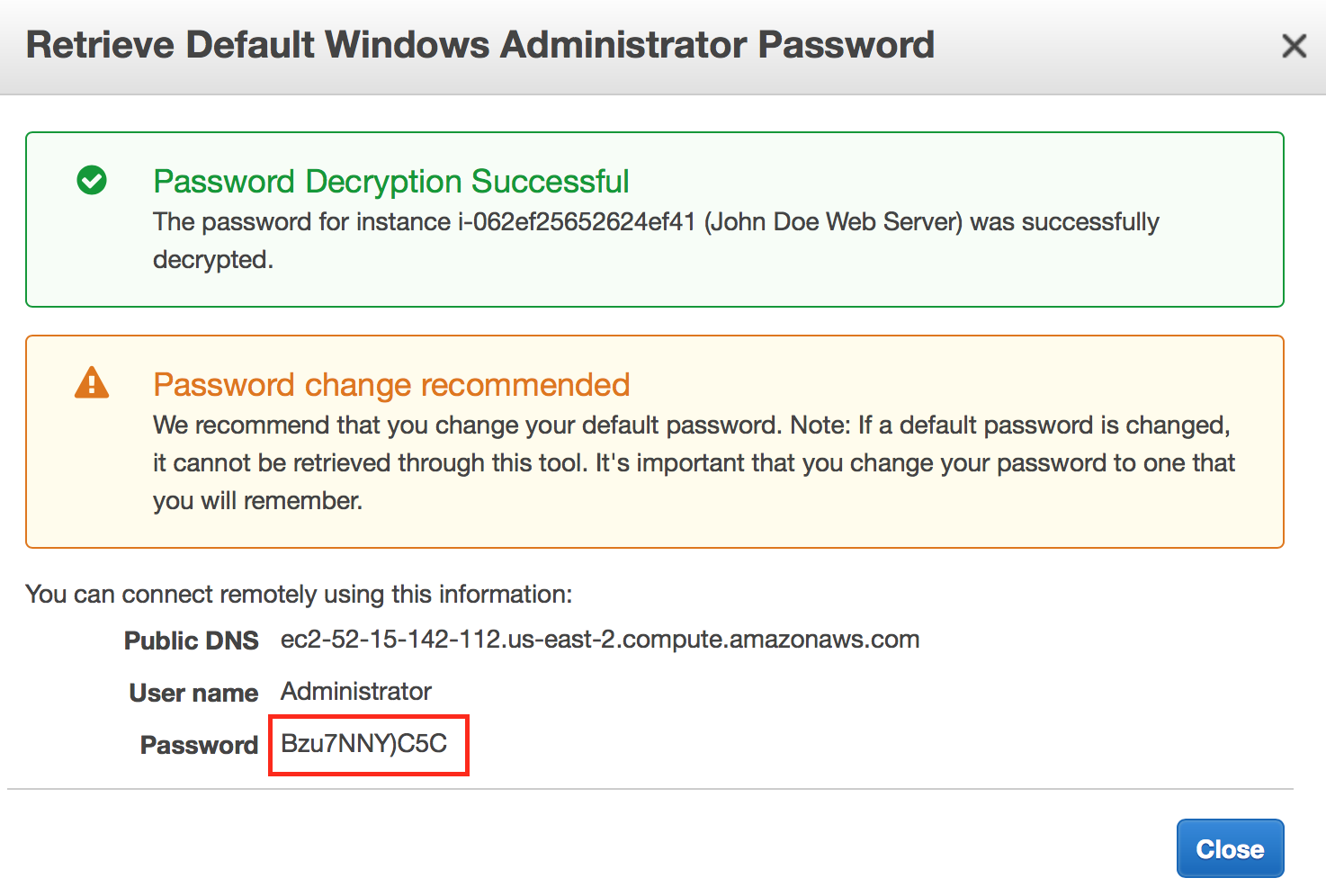
1. Retrieve the automatically generated, encrypted Windows password by selecting **Get Windows Password** from the Action menu (or by right clicking your instance).



1. On the next screen, click the Choose File button and select the .pem private key file that was downloaded earlier when you launched the instance. Then click **Decrypt Password** to obtain the Administrator password.



1. The decrypted Administrator password should look something like this:

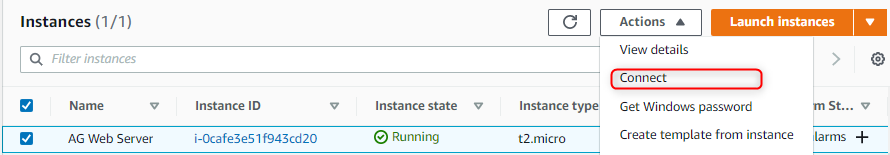


Copy the password to clipboard and save it for the future use.

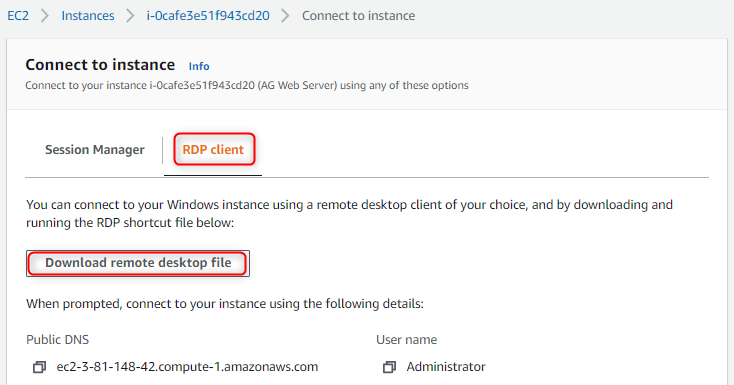
**Note** that since only you have the private key, it’s important to understand the automatically generated password can only be decrypted by you. So it’s important to keep this key secure. Generally, the automatically generated password is changed by the customer after first login. If the automatically generated password is not changed and the private key is lost, there’s no way to recover the password.

1. Start your RDP application and **connect** to the hostname of your instance.

Choose **RDP Client** instead of the **Session Manager**.



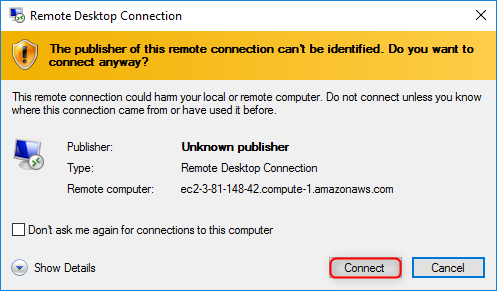
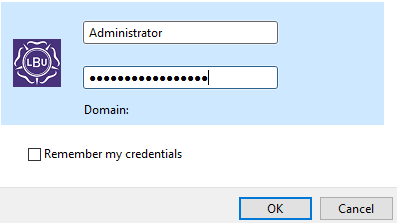
Download the **.rdp** file when prompted and open it with RDP app (simply double-click on the downloaded file).



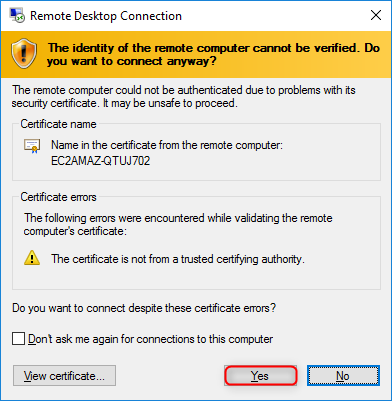
Amazon EC2 instances created from most Windows Amazon Machine Images (AMIs) enable you to connect using **RDP Client**. RDP Client uses the Remote Desktop Protocol (**RDP**) and enables you to connect to and use your instance in the same way you use a computer sitting in front of you.

**Session Manager** is a fully managed AWS Systems Manager capability that lets you manage your Amazon EC2 instances, on-premises instances, and virtual machines (VMs) through an interactive one-click browser-based shell or through the AWS CLI.

1. In your RDP application, use ***Administrator*** as the username along with the decrypted password. Once connected, you will have access to the Windows desktop. At this point, feel free to explore Windows. If you would like, you can change the Administrator password to something friendlier or easy to remember (but still secure of course).

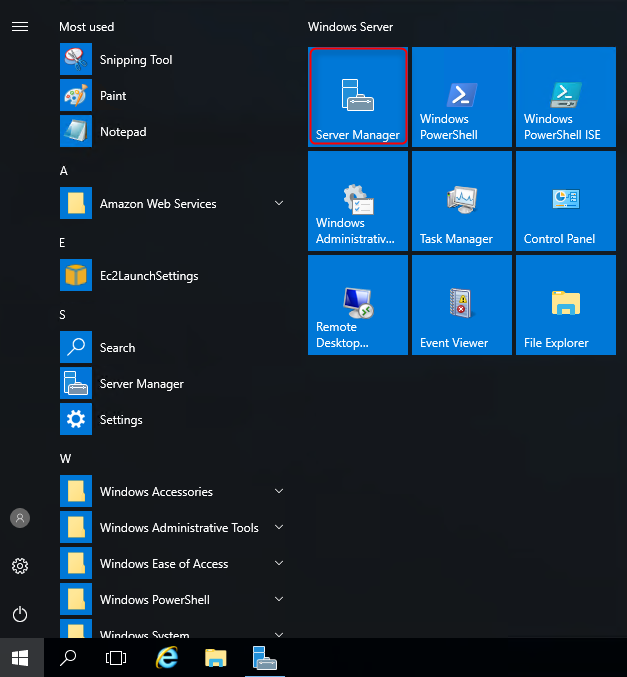
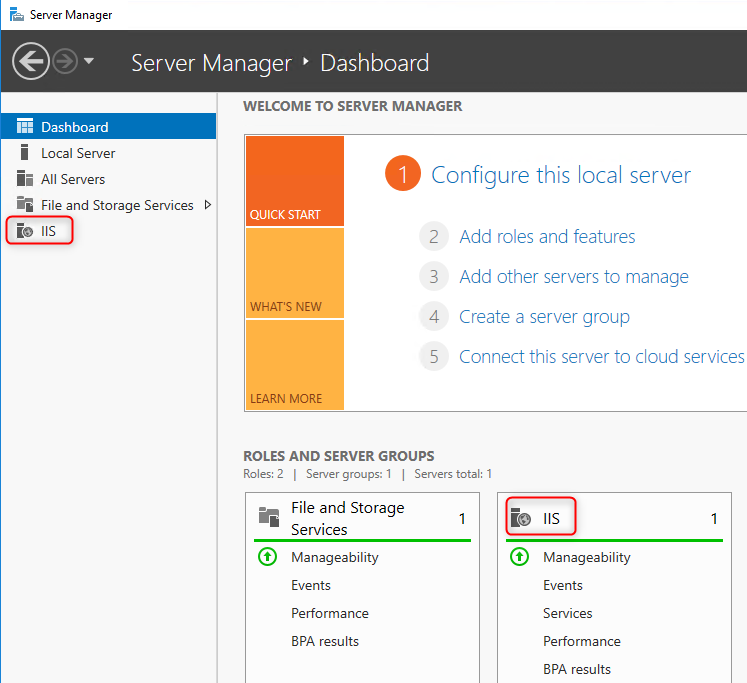
 

Click OK, then accept connection despite certificate issue by clicking Yes.

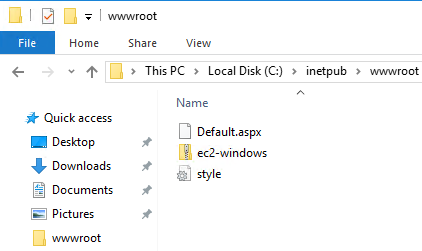


Now you are logged into your Windows server VM.

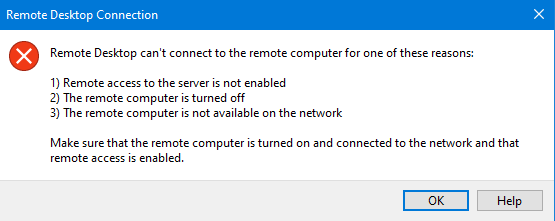
You can open Server Manager to make sure IIS web server is run.

If you check the **wwwroot** folder of the IIS you could see files uploaded there by user powershell script we created at launching the VM.



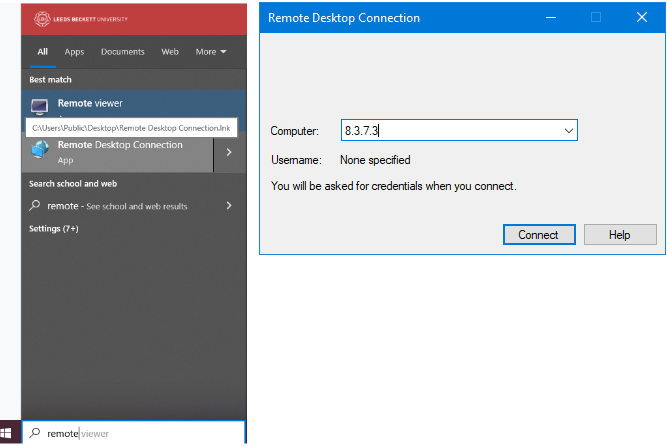
**If you get the connection error.**



This might be caused by misconfiguration between the .rdp file you downloaded from AWS and remote desktop settings on university PCs.

To fix it,

1. run the Remote Desktop (NOT Remote Viewer) app from your university PC and copy/paste the DNS name or public(!) IP address of your Windows VM and click ‘Connect’.



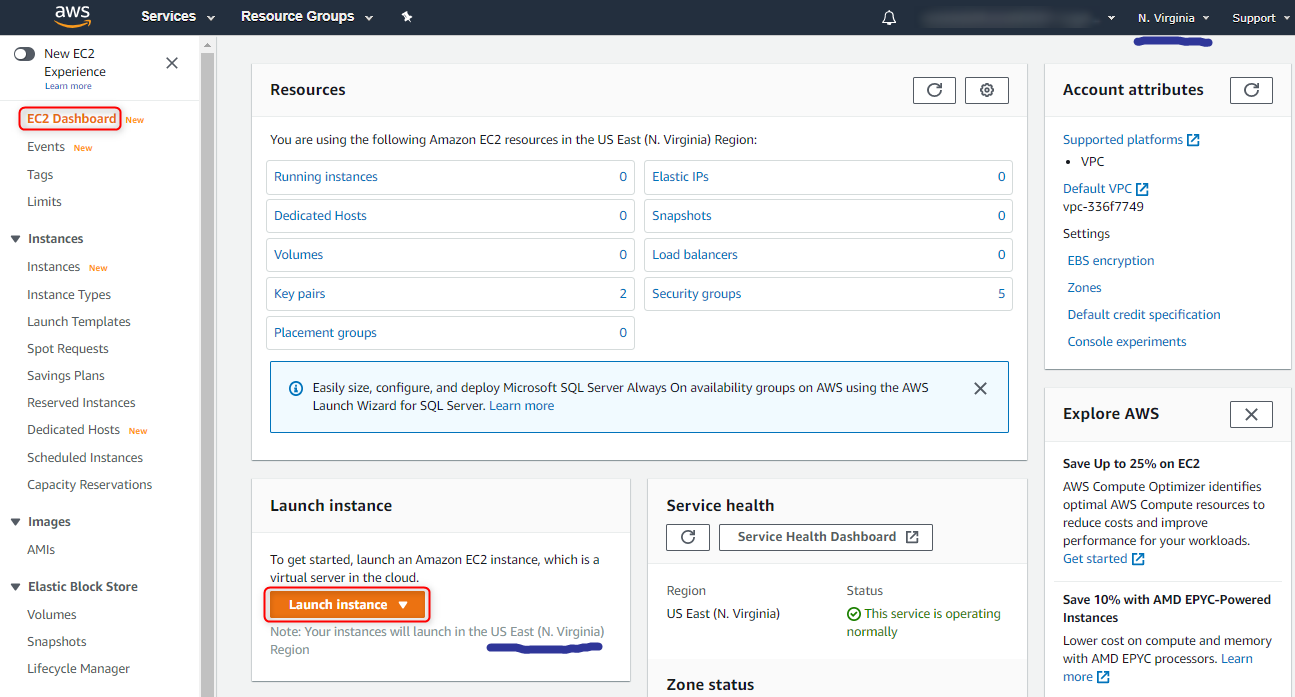
1. The suggested user name will be university id. Instead, select ‘Another user.’
2. As the user name type ‘\Administrator’ (‘\’ is important to switch from the default ‘LEEDSBECKETT’ domain to the blank domain).
3. Decrypt Administrator’s password from the .pem file as instructed above and use it to log in.

# Step 4: Running Linux Server VM on Amazon EC2

## Launch a Linux Server Virtual Instance

In this example we will launch a Linux instance with the web server installed upon boot.

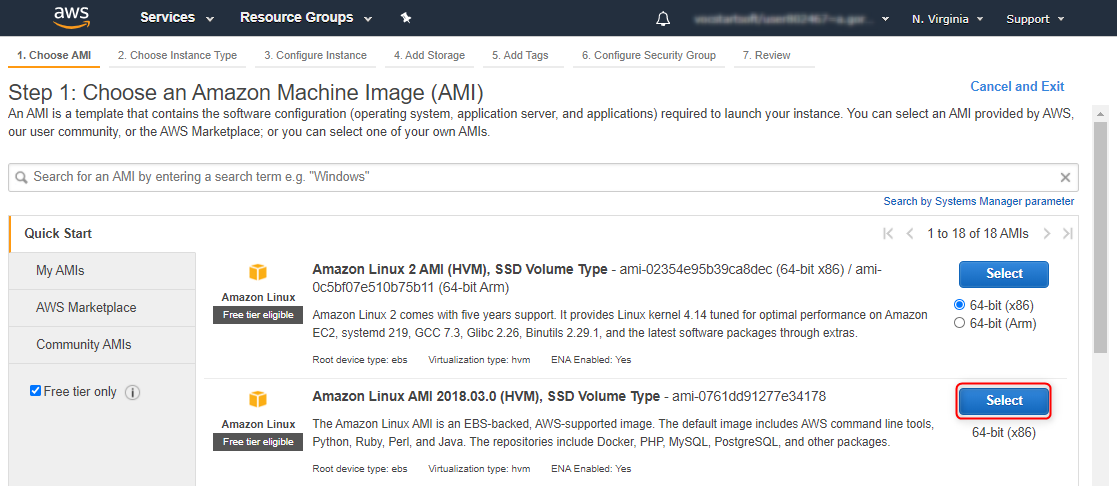
1. Once you opened the Amazon EC2 console in the AWS Management Console, click on **Launch Instance**



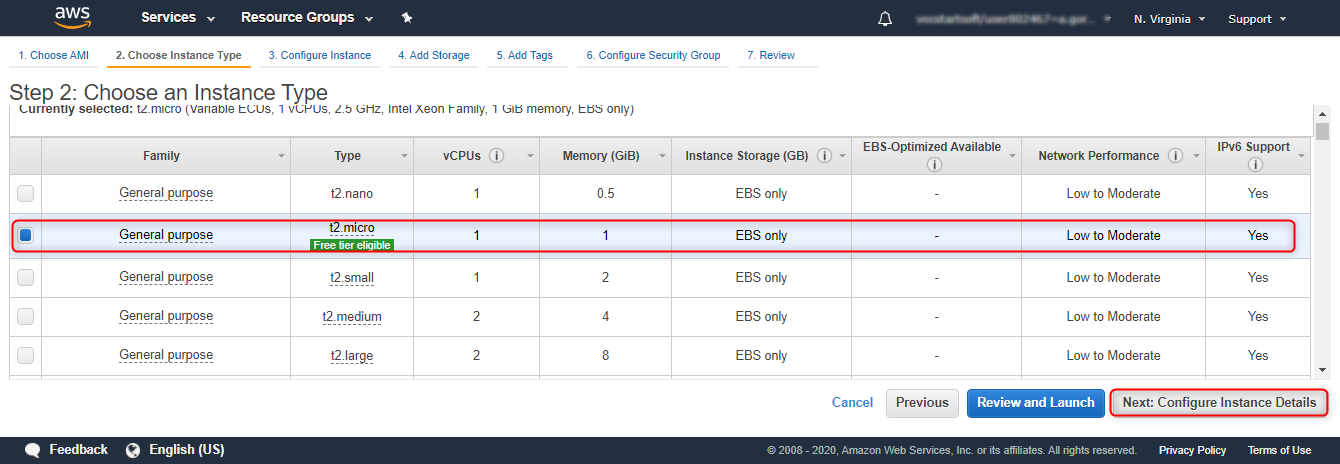
On the left panel pick out “**Free tier only**” to display only those VM images which are eligible for the free tier usage.



1. Scroll down and click **Select** on some Linux instance template, e,g, the **Amazon Linux 2 AMI**.



1. In the **Choose Instance Type** tab, select the **t2.micro** instance size and click **Next: Configure Instance Details**



1. On the **Configure Instance Details** page, expand the **Advanced Details** section, copy/paste the script into the User Data field (when copy/paste from Google Drive some special characters, like ‘<’, ‘>’, etc. could be replaced with their codes; thus, it is recommended to download the labsheet on your PC and open in MS Word from where you can copy/paste the code):



#!/bin/sh

yum -y install httpd php mysql php-mysql

case $(ps -p 1 -o comm | tail -1) in

systemd) systemctl enable --now httpd ;;

init) chkconfig httpd on; service httpd start ;;

\*) echo "Error starting httpd (OS not using init or systemd)." 2>&1

esac

if [ ! -f /var/www/html/bootcamp-app.tar.gz ]; then

cd /var/www/html

wget https://github.com/anatoliy-gorbenko/CloudComputing/blob/main/bootcamp-app.tar?raw=true

tar xvf bootcamp-app.tar

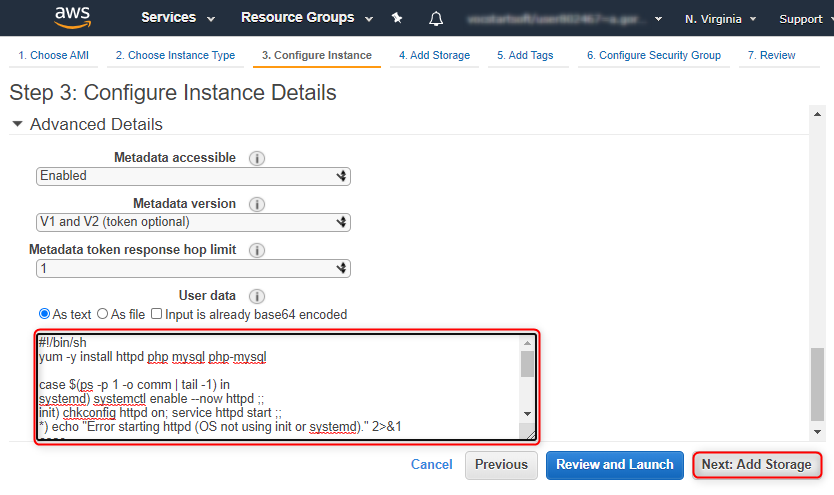
chown apache:root /var/www/html/rds.conf.php

fi

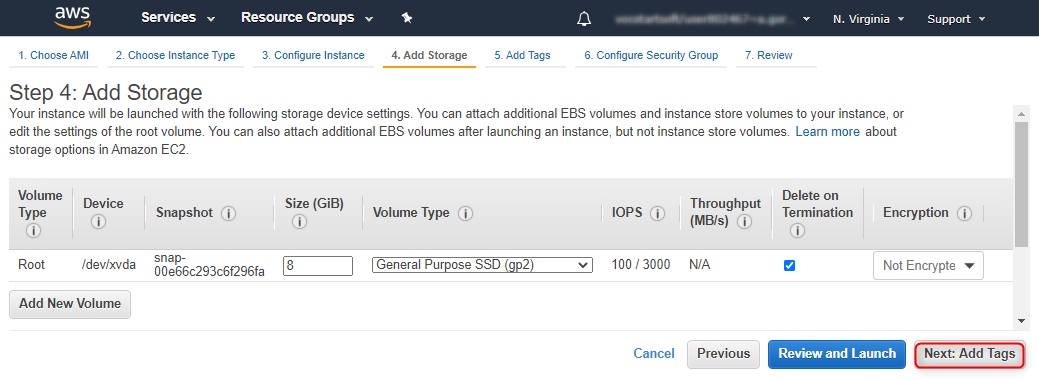
yum -y update

This Shell script will install and start the LAMP stack (httpd, php, mysql, php-mysql) and deploy a simple web page by downloading and unzipping bootcamp.tar archive with the sample .php pages into a /var/www/html folder of the apache web server.

For further information on User Data please refer to the documentation at - <http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ec2-instance-metadata.html>.



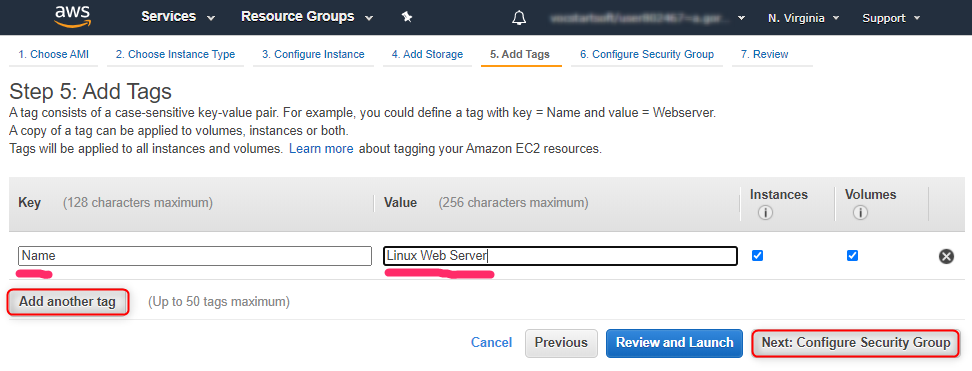
1. Click **Next: Add Storage**:



On the **Step 4: Add Storage** screen, Click **Next: Add Tags** to accept the default Storage Device Configuration and move to the **Step 5: Add Tags screen**.

Next, choose a “friendly name” for your instance. This name, more correctly known as a tag, will appear in the console once the instance launches. It makes it easy to keep track of running machines in a complex environment.

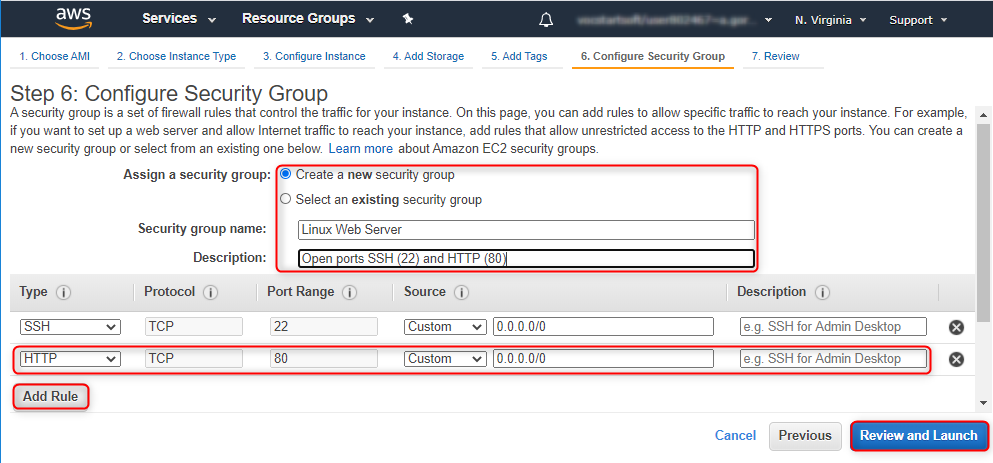
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = “**Name**”and value = “**[Your Name] Linux Web Server**”.



Then click **Next: Configure Security Group**.

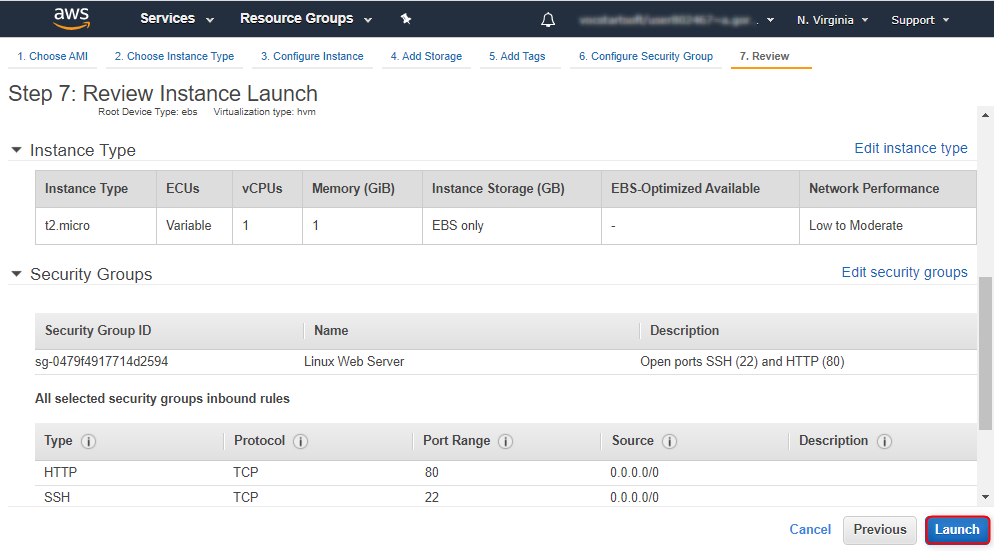
1. You will be prompted to create a new security group, which will be your firewall rules. On the assumption that we are building out a Web server, name this security group according to this format “**[Your Name] Linux Web Server**”.

**Add Rule** to open port 80 (HTTP) in addition to 22 (SSH). Click the **Review and Launch** button after configuring the security group.

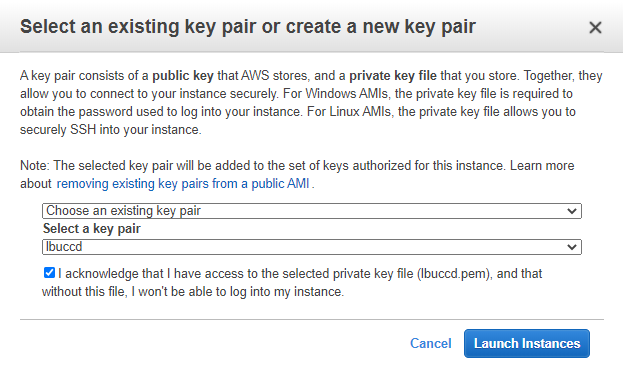


1. Review your choices, and then click **Launch**.

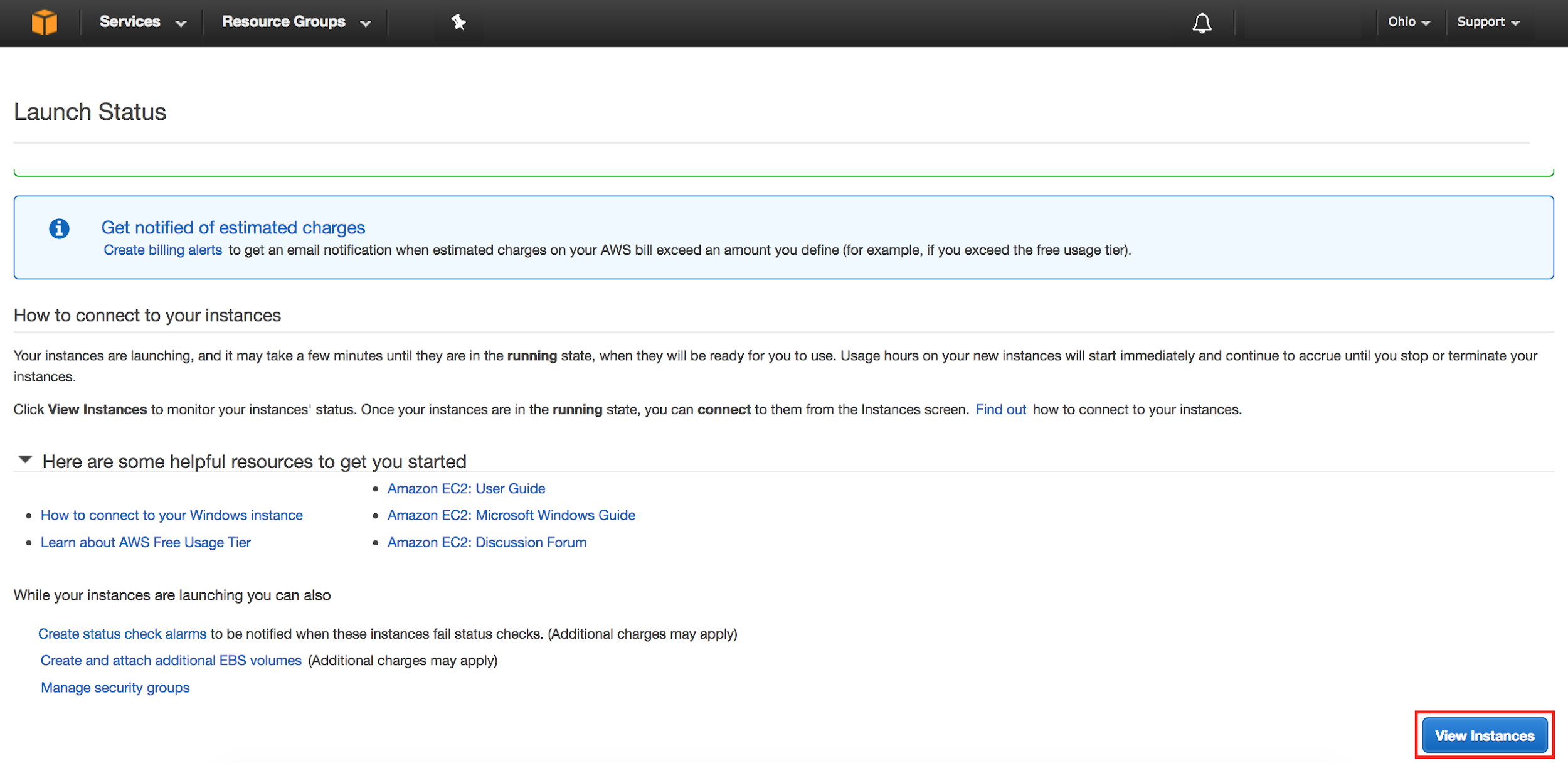
***Note*** *the warning boxe(s) at the top of the page, this is to warn you about possible configuration issues. In this lab we are creating a Windows server that has RDP and HTTP access that is “open to the world” this is something that you wouldn’t normally do.*



1. Now you need to select an existing public/private keypair (you can use one created when you lounched Windows VM) or to create a new one.



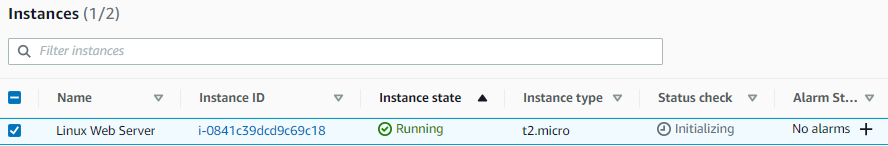
1. Now click the **Launch Instances** button to launch your linux web server.
2. The next screen will confirm that your instance is now launching. Click the **View Instances** button. Once your instance has launched, you will see the “**[Your Name] Linux Web Server**” instance as well as the Availability Zone the instance is in and its publicly routable DNS name.



## Browse the Web Server

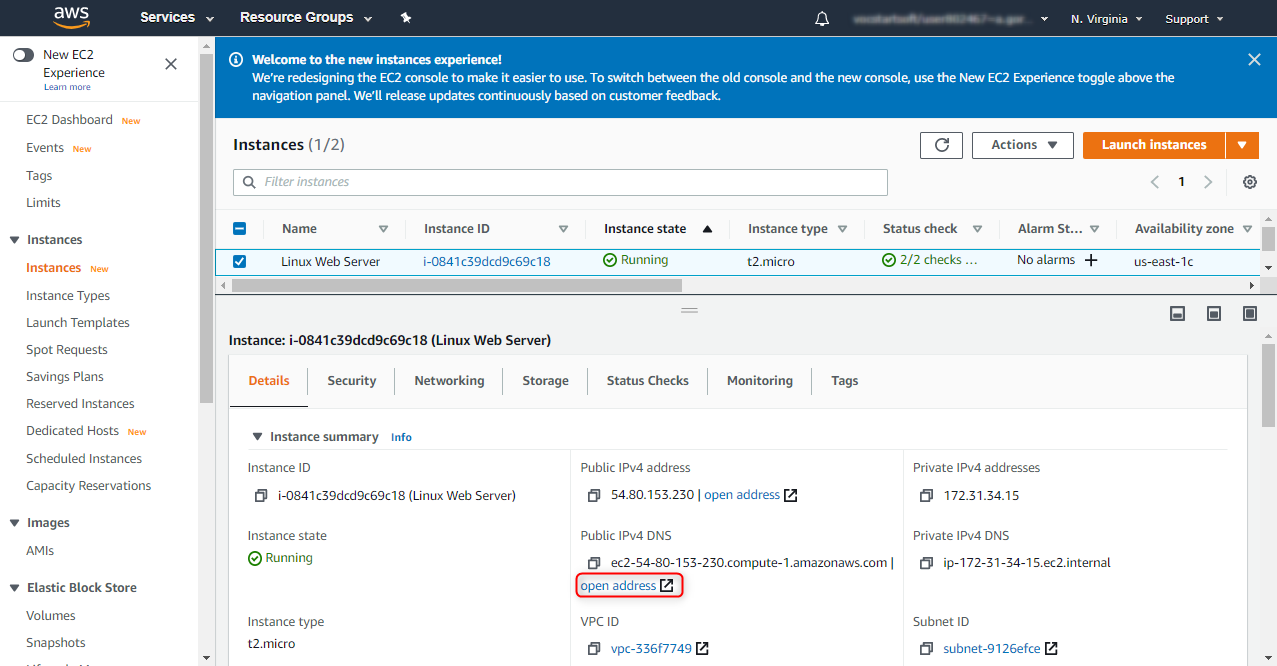
Now you will browse to the Web Server site that was installed on the Instance using the PowerShell script defined in the **User Data** section during creation of the instance.

1. Wait for the instance to pass the Status Checks. For linux instances, this could take up to 10 minutes.



When complete, you will see the Status Checks have passed.

From the window below you can also notice your VM’s IP address and Public DNS name.



1. Click on **Open address** or open a new browser tab and browse the Web Server by entering the EC2 instance’s Public DNS name into the browser (make sure you use ’http://’ prefix, not ‘https://’). You should see a page that looks similar to this (if not, wait for some time and reload the page):





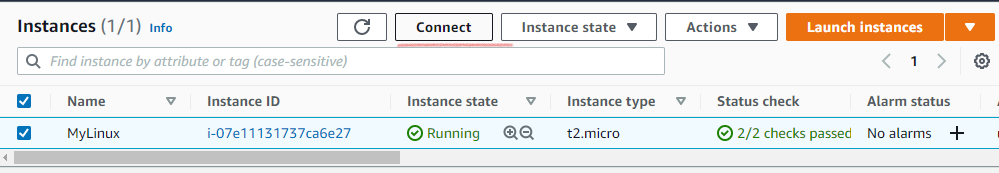
# Step 5: Connecting to Your EC2 Linux Instance via SSH

There are a few ways to connect to your Linux Instance on Amazon EC2. You can use:

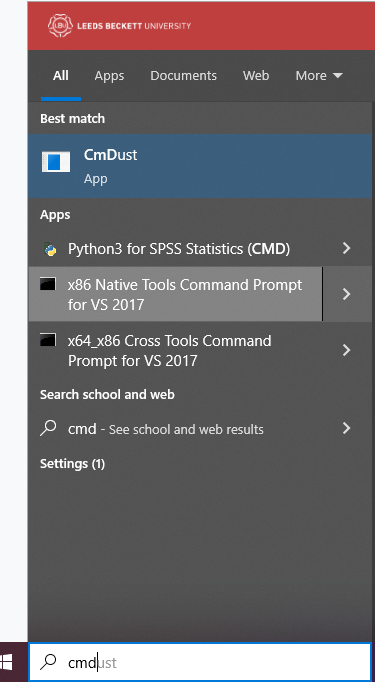
* **Amazon EC2 Instance Connect** provides a simple and secure way to connect to your instances using using a browser-based SSH client. Learn more at <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-connect-set-up.html>
* **Session Manager** is a fully managed AWS Systems Manager capability that lets you manage your Amazon EC2 instances, on-premises instances, and virtual machines (VMs) through an interactive one-click browser-based shell or through the AWS CLI. Learn more at <https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager.html>
* **SSH client** provides a simple and secure way to connect to your instances using Secure Shell (SSH) from any Linux/MacOS or even from Windows machine (using third-party’s tools, e.g. Putty <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html> ). Learn more at <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html>.

SSH client is a part of Linux / MacOS. It has been recently incorporated into Windows as well.

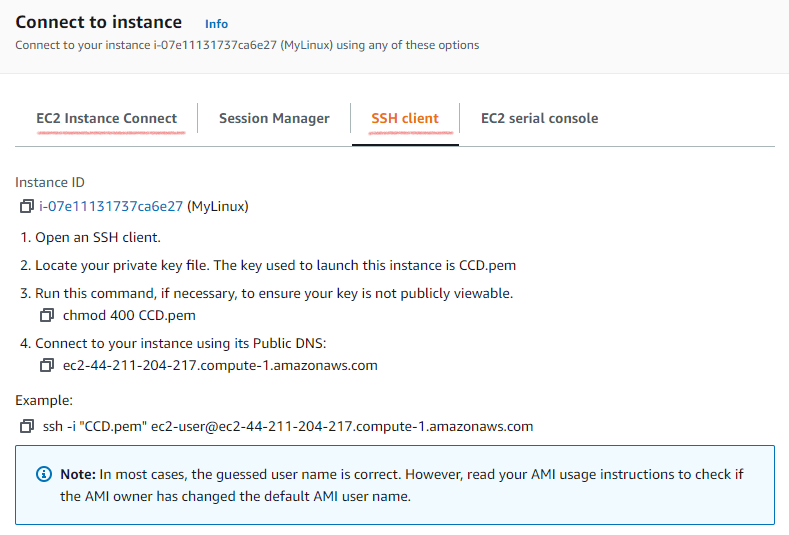
Make sure your Linux instance is running, select it and click on ‘Connect’.



Here you will see a variety of options to connect to your virtual Linux instance. The easiest way is to use the first option ‘EC2 Instance Connect’. This will open the in-browser shell window. Also explore the ‘SSH client’ connection option making use of ssh client available on windows (skip step 3 if on Windows).

For this purpose, 

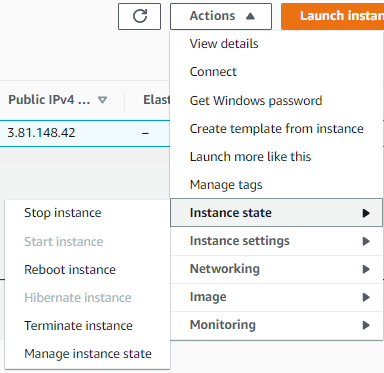
1. open the cmd window (NOT CmDust)
2. locate the key-pair .pem file you downloaded earlier (it might be easier to navigate to the same folder using ‘cd’ command, e.g. ‘cd c:\users\c123456\downloads’). - use your user id instead of ‘c123456’.
3. establish ssh connection using the example link provided, e.g. ‘ssh -i "CCD.pem" ec2-user@ec2-44-217.compute-1.amazonaws.com’



After you establish the ssh connection, try a few linux commands, e.g. ‘whoami’, ‘pwd’, ‘ls’.

# Step 6: Terminate instances after use!!!





# References

IaaS PaaS on Amazon AWS EC2 <https://www.youtube.com/watch?v=H6QEGQBlWXs&t=1161s>

EC2 Key Pair Creation <https://techexpert.tips/amazon-aws/ec2-key-pair-creation/>

EC2 Ubuntu Linux Virtual Machine <https://techexpert.tips/amazon-aws/ec2-ubuntu-linux-virtual-machine/>

EC2 Windows Virtual Machine <https://techexpert.tips/amazon-aws/ec2-windows-virtual-machine/>

AWS - IaaS Compute / EC2 Instance <https://www.youtube.com/watch?v=t_DwdPdq_5c&t=1454s>

Tutorial: Getting started with Amazon EC2 Linux instances <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html>

Tutorial: Getting started with Amazon EC2 Windows instances <https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/EC2_GetStarted.html>

How to Set Up IIS on an EC2 Windows Server 2016 Instance <https://www.youtube.com/watch?v=OPHT0ozZOj4>

How to Use a GUI with Ubuntu Linux on AWS EC2 <https://www.youtube.com/watch?v=6x_okhl_CF4&t=3s>