



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
Step 1: Logging in the AWS Management Console	3
Step 2: Running Windows Server VM on Amazon EC2	6
Launch a Windows Server Virtual Instance	6
Browse the Web Server	13
Step 3: Connecting To Your Windows Instance	15
Step 4: Running Linux Server VM on Amazon EC2	20
Launch a Linux Server Virtual Instance	20
Browse the Web Server	26
Step 5: Connecting to Your EC2 Linux Instance	28
Converting .pem private key to .ppk private key used by PuTTY	28
Connecting to Linux VM using PuTTY	29
Step 6: Terminate instances after use!!!	31
References	32

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](#) 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. .

Course/Lab	Category	Duration	Progress
Getting Started with Storage (Lab)	Cloud Computing	Foundational 2 hour(s)	0%
Getting Started with Compute (Lab)	Cloud Computing	Foundational 2 hour(s)	60%
Introduction to the AWS Management Console	Cloud Computing	Foundational 1 hour(s)	0%
Builder Labs	Development	Intermediate 6 hour(s)	0%
Introduction to Cloud 101 (Labs)	Cloud Computing	Foundational 8 hour(s)	0%

The screenshot shows the AWS Management Console homepage. At the top, there is a navigation bar with icons for back, forward, and home, followed by the URL "console.aws.amazon.com/console/home?region=us-east-1#". Below the URL is the AWS logo and a "Services" dropdown menu. The main title "AWS Management Console" is centered above the "AWS services" section. The "AWS services" section includes a "Find Services" search bar with placeholder text "Example: Relational Database Service, database, RDS". Below the search bar is a "Recently visited services" section with links to Elastic Beanstalk, IAM, Lambda, and API Gateway. The "All services" section is expanded, showing categories like Compute, Blockchain, Security, Identity, & Compliance, and others. The "Compute" category is expanded further, showing EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, and EC2 Image Builder. The "EC2" link is highlighted with a red box.

AWS services

Find Services
You can enter names, keywords or acronyms.
Example: Relational Database Service, database, RDS

▼ Recently visited services

Elastic Beanstalk IAM Billing
Lambda API Gateway

▼ All services

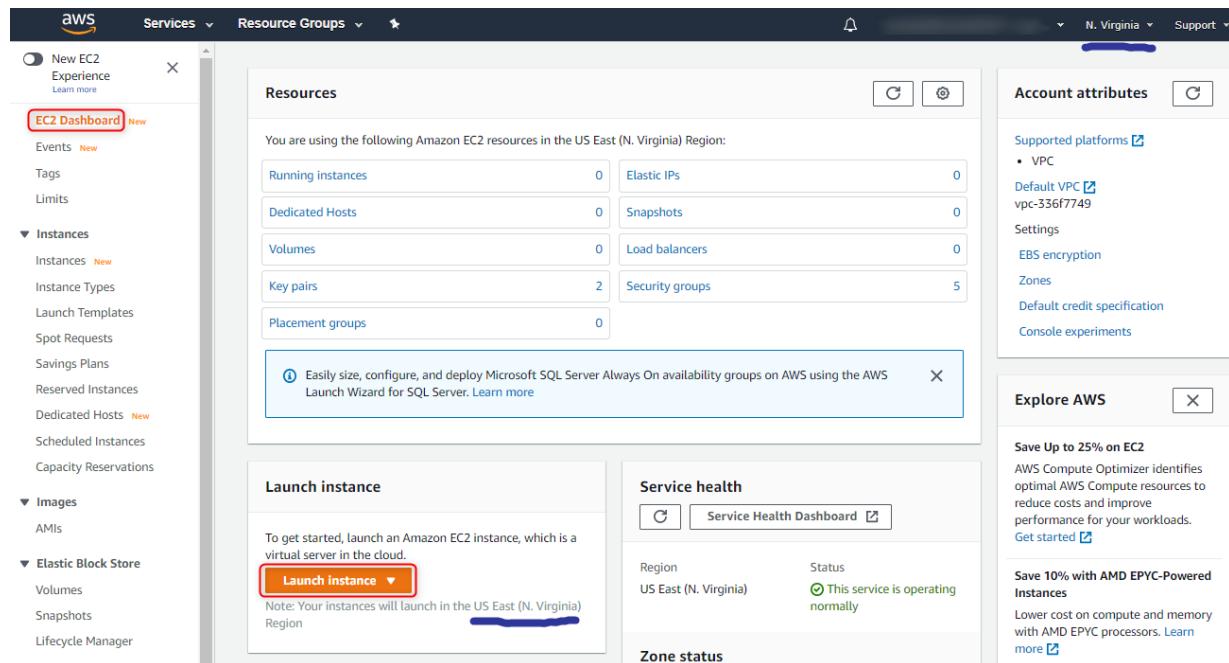
<input type="checkbox"/> Compute EC2 Lightsail Lambda Batch Elastic Beanstalk Serverless Application Repository AWS Outposts EC2 Image Builder	<input type="checkbox"/> Blockchain Amazon Managed Blockchain <input type="checkbox"/> Satellite Ground Station <input type="checkbox"/> Quantum Technologies Amazon Braket	<input type="checkbox"/> Security, Identity, & Compliance IAM Resource Access Manager Cognito Secrets Manager GuardDuty Inspector Amazon Macie
--	--	---

Step 2: Running Windows Server VM on Amazon EC2

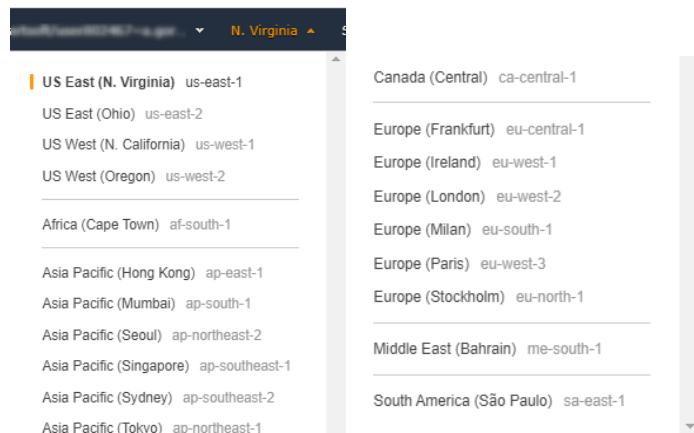
Launch a Windows Server Virtual Instance

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

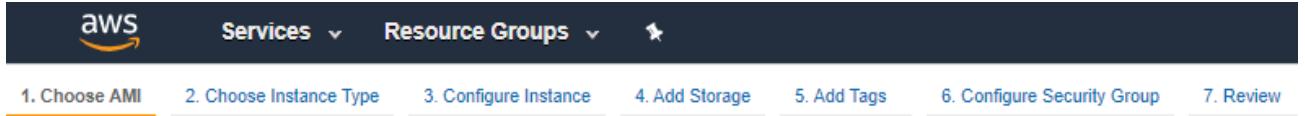
- Once you opened the Amazon EC2 console in the AWS Management Console, click on **Launch Instance**



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.



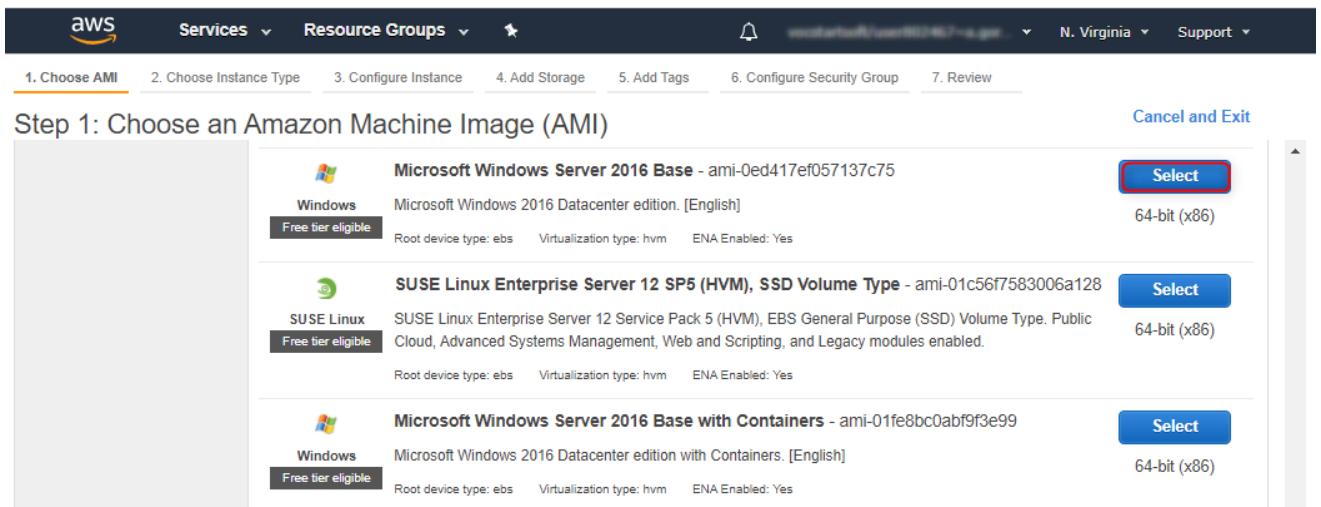
On the top of your screen you can notice the general workflow of creating a VM instance. It includes Choosing Amazon Machine Image (AMI), Choosing Instance Type, Configuring the Instance, Adding Storage, Adding Tags, Configuring Security Groups and Reviewing:



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



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



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

The screenshot shows the AWS CloudFormation console. The top navigation bar includes 'Services', 'Resource Groups', and tabs for '1. Choose AMI', '2. Choose Instance Type' (which is active), '3. Configure Instance', '4. Add Storage', '5. Add Tags', '6. Configure Security Group', and '7. Review'. Below the tabs, a section titled 'Step 2: Choose an Instance Type' displays the message 'Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)'. A table lists various instance types:

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	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 <small>Free tier eligible</small>	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

At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Configure Instance Details'.

5. 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:



'User data' is a method for bootstrapping your instance - Any code placed here will be executed the first time an instance is launched.

```
<powershell>
Install-WindowsFeature Web-Server -IncludeManagementTools -IncludeAllSubFeature
remove-item -recurse c:\inetpub\wwwroot\*
(New-Object
System.Net.WebClient).DownloadFile("https://gitlab.com/karpenkoKhAI/ec2-
template/-/raw/main/ec2-windows.zip", "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>.

Metadata accessible [Info](#)

Select

Metadata transport

Select

Metadata version [Info](#)

Select

Metadata response hop limit [Info](#)

Select

Allow tags in metadata [Info](#)

Select

User data - *optional* [Info](#)
Upload a file with your user data or enter it in the field.

```
<powershell>
Install-WindowsFeature Web-Server -IncludeManagementTools -
IncludeAllSubFeature
remove-item -recurse c:\inetpub\wwwroot\*
(New-Object
System.Net.WebClient).DownloadFile("https://gitlab.com/karpenkoKhAI/ec2-
template/-/raw/main/ec2-windows.zip", "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>
```

User data has already been base64 encoded

6. 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**.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	John Doe Web Server

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch 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.

7. 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.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security group name: John Doe Web Server

Description: This rule allows RDP & HTTP access to John Doe Web Server

Type	Protocol	Port Range	Source
RDP	TCP	3389	Custom 0.0.0.0/0
HTTP	TCP	80	Custom 0.0.0.0/0

Add Rule

Warning
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.

Cancel Previous Review and Launch

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

Note the warning box(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.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Microsoft Windows Server 2016 Base - ami-0ed417ef057137c75

Free tier eligible Microsoft Windows 2016 Datacenter edition. [English]
Root Device Type: ebs Virtualization type: hvm

If you plan to use this AMI for an application that benefits from Microsoft License Mobility, fill out the [License Mobility Form](#). Don't show me this again

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

AG Web Server
The rule allows RDP and HTTP

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	

Instance Details

Storage

Cancel **Previous** **Launch**

- 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.

Select an existing key pair or create a new key pair

X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Select a key pair



No key pairs found

You don't have any key pairs. Please create a new key pair by selecting the [Create a new key pair](#) option above to continue.

[Cancel](#)
[Launch Instances](#)

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

Select an existing key pair or create a new key pair

X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Key pair name

[Download Key Pair](#)


You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

[Cancel](#)
[Launch Instances](#)

11. 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.

12. Now click the **Launch Instances** button to launch your Windows web server.

13. 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.

The screenshot shows the AWS Launch Status page. At the top, there are navigation links for Services, Resource Groups, and a search bar. On the right, there are links for Ohio, Support, and a bell icon. Below the header, the title "Launch Status" is displayed. A callout box provides information about estimated charges and billing alerts. The main content area includes sections for "How to connect to your instances" and "Helpful resources". Under "Helpful resources", a section titled "Here are some helpful resources to get you started" lists links to the Amazon EC2 User Guide, How to connect to your Windows instance, Learn about AWS Free Usage Tier, and the Amazon EC2 Discussion Forum. Below these, there's a note about launching instances and a "View Instances" button, which is highlighted with a red box.

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.

The screenshot shows the AWS Instances page. The table header includes columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, and Alarm Status. A row for the "John Doe Web Server" instance is shown, with its details: Name (John Doe Web Server), Instance ID (i-062ef25652624ef41), Instance Type (t2.medium), Availability Zone (us-east-2c), Instance State (pending), Status Checks (Initializing), and Alarm Status (None). The "Status Checks" column is highlighted with a red box.

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.

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm St...
AG Web Server	i-0cafe3e51f943cd20	Running	t2.micro	2/2 checks ...	No alarms +

Instance: i-0cafe3e51f943cd20 (AG Web Server)

Details Security Networking Storage Status Checks Monitoring Tags

Instance summary

Instance ID i-0cafe3e51f943cd20 (AG Web Server)	Public IPv4 address 3.81.148.42 open address	Private IPv4 addresses 172.31.44.199
Instance state Running	Public IPv4 DNS ec2-3-81-148-42.compute-1.amazonaws.com open address	Private IPv4 DNS ip-172-31-44-199.ec2.internal
Instance type t2.micro	VPC ID vpc-336f7749	Subnet ID subnet-9126efce

- 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):

Not secure | ec2-3-81-148-42.compute-1.amazonaws.com

Welcome to the Cloud Computing Development Module!

Meta-Data	Value
Instance Id	i-0cafe3e51f943cd20
Availability Zone	us-east-1c

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](#). If your home PC's is Linux you can use one of the following RDP clients: [Remmina](#), [FreeRDP](#), [rdesktop](#) or [Vinagre](#).

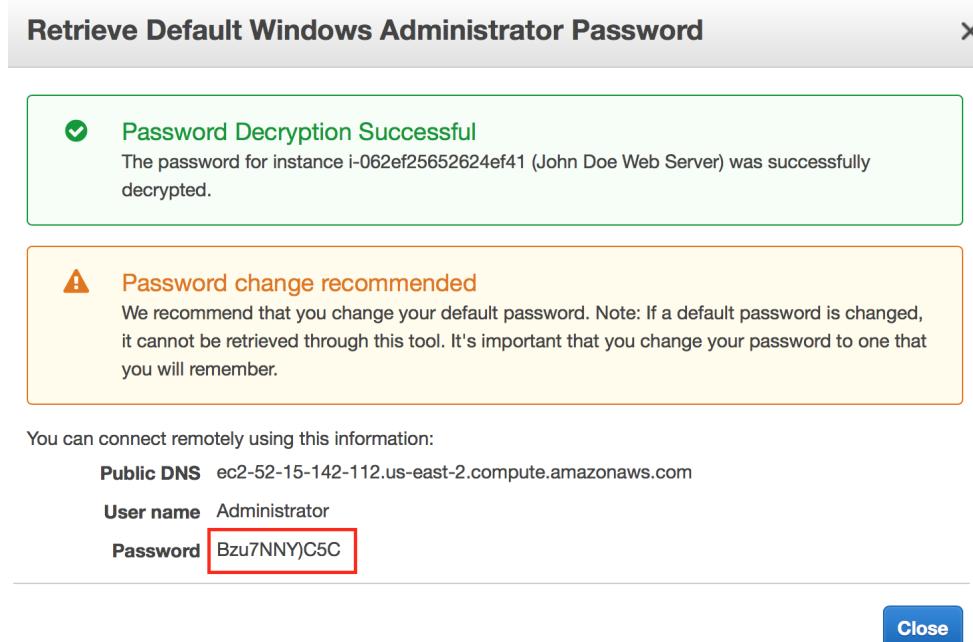
1. Retrieve the automatically generated, encrypted Windows password by selecting **Get Windows Password** from the Action menu (or by right clicking your instance).

The screenshot shows the AWS CloudWatch Instances interface. In the top right corner, there is an 'Actions' button with a dropdown arrow. Below it, the 'Get Windows password' option is highlighted with a red box. Other options like 'View details', 'Connect', and 'Create template from instance' are also visible.

2. 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.

The dialog box has a title 'Retrieve Default Windows Administrator Password'. It contains instructions: 'To access this instance remotely (e.g. Remote Desktop Connection), you will need your Windows Administrator password. A default password was created when the instance was launched and is available encrypted in the system log.' Below this, it says: 'To decrypt your password, you will need your key pair for this instance. Browse to your key pair, or copy and paste the contents of your private key file into the text area below, then click Decrypt Password.' It shows a 'Key Name' field with 'Vinod-Ohio-Keypair' and a 'Key Pair Path' field with 'Choose File' and 'JohnDoe-Keypair.pem.txt'. It also says: 'Or you can copy and paste the contents of the Key Pair below:' followed by a large text area containing an RSA private key. At the bottom right are 'Cancel' and 'Decrypt Password' buttons, with 'Decrypt Password' also highlighted with a red box.

3. 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.

4. Start your RDP application and **connect** to the hostname of your instance. Choose **RDP Client** instead of the **Session Manager**.

The screenshot shows the AWS CloudWatch Instances dashboard. The top navigation bar includes "Instances (1/1)", "Actions ▾", "Launch instances ▾", and a dropdown menu. Below the navigation is a search bar labeled "Filter instances". The main table displays one instance: "AG Web Server" (Instance ID: i-0cafe3e51f943cd20, Instance state: Running, Instance type: t2.micro). To the right of the table are several actions: "View details", "Connect" (which is highlighted with a red box), "Get Windows password", "Edit St...", "Create template from instance", and "Alarms +".

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

EC2 > Instances > i-0cafe3e51f943cd20 > Connect to instance

Connect to instance Info

Connect to your instance i-0cafe3e51f943cd20 (AG Web Server) using any of these options

Session Manager | **RDP client**

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

Download remote desktop file

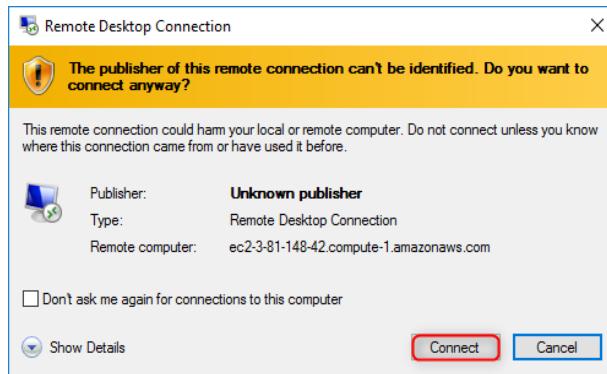
When prompted, connect to your instance using the following details:

Public DNS	User name
<input type="text"/> ec2-3-81-148-42.compute-1.amazonaws.com	<input type="text"/> Administrator

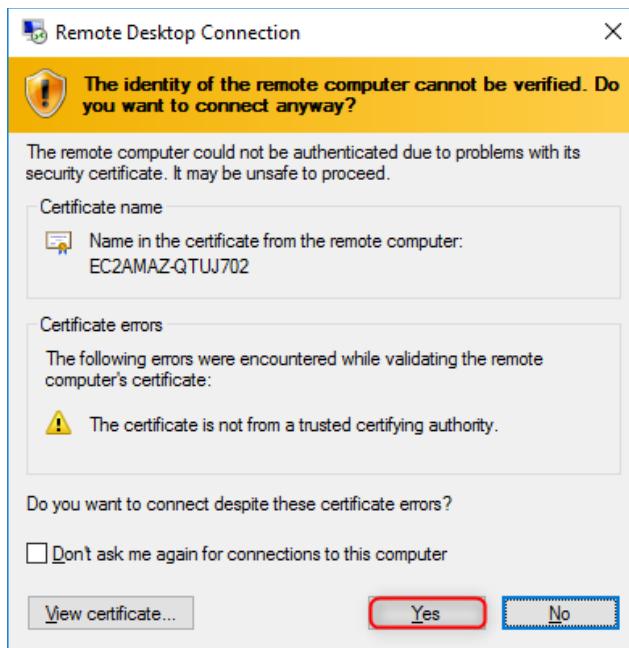
Amazon EC2 instances created from most Windows Amazon Machine Images (AMIs) enable you to connect using **RDP Client**. RDP Client uses the Remote Desktop Protocol (**RDP**) and enables you to connect to and use your instance in the same way you use a computer sitting in front of you.

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.

5. 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).



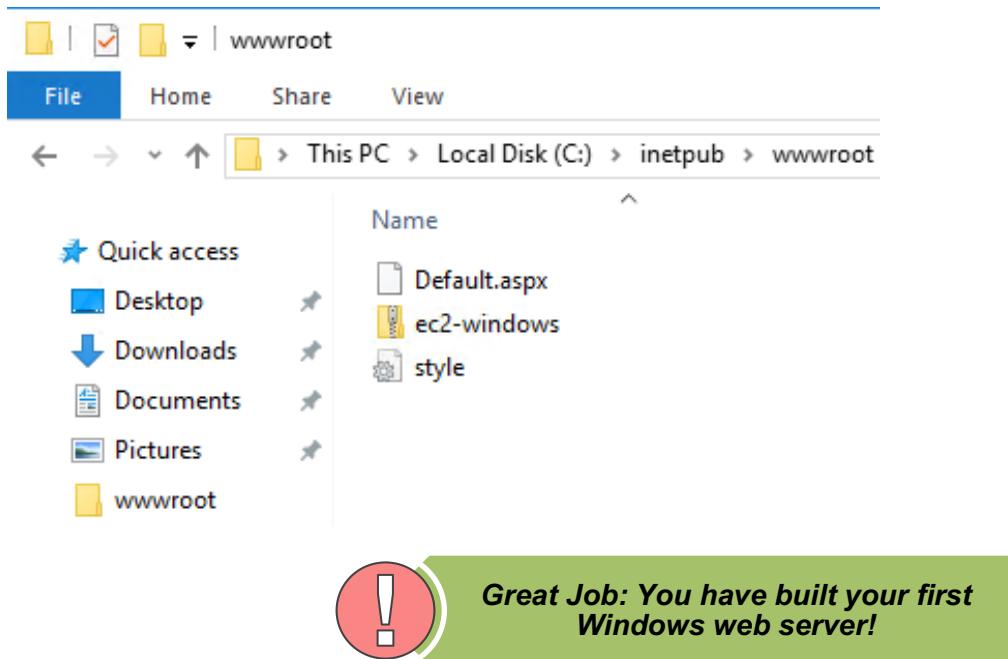
Click OK, then accept connection despite certificate issue by clicking Yes.



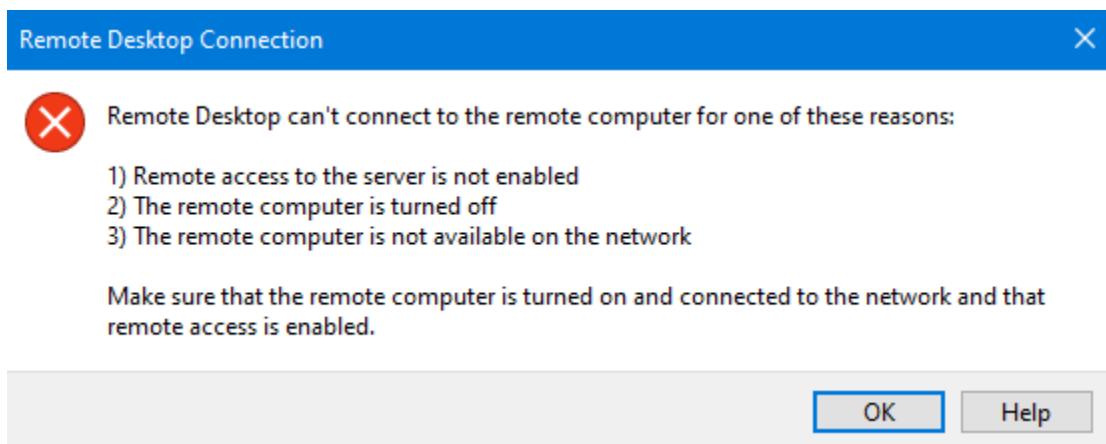
Now you are logged into your Windows server VM.

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.



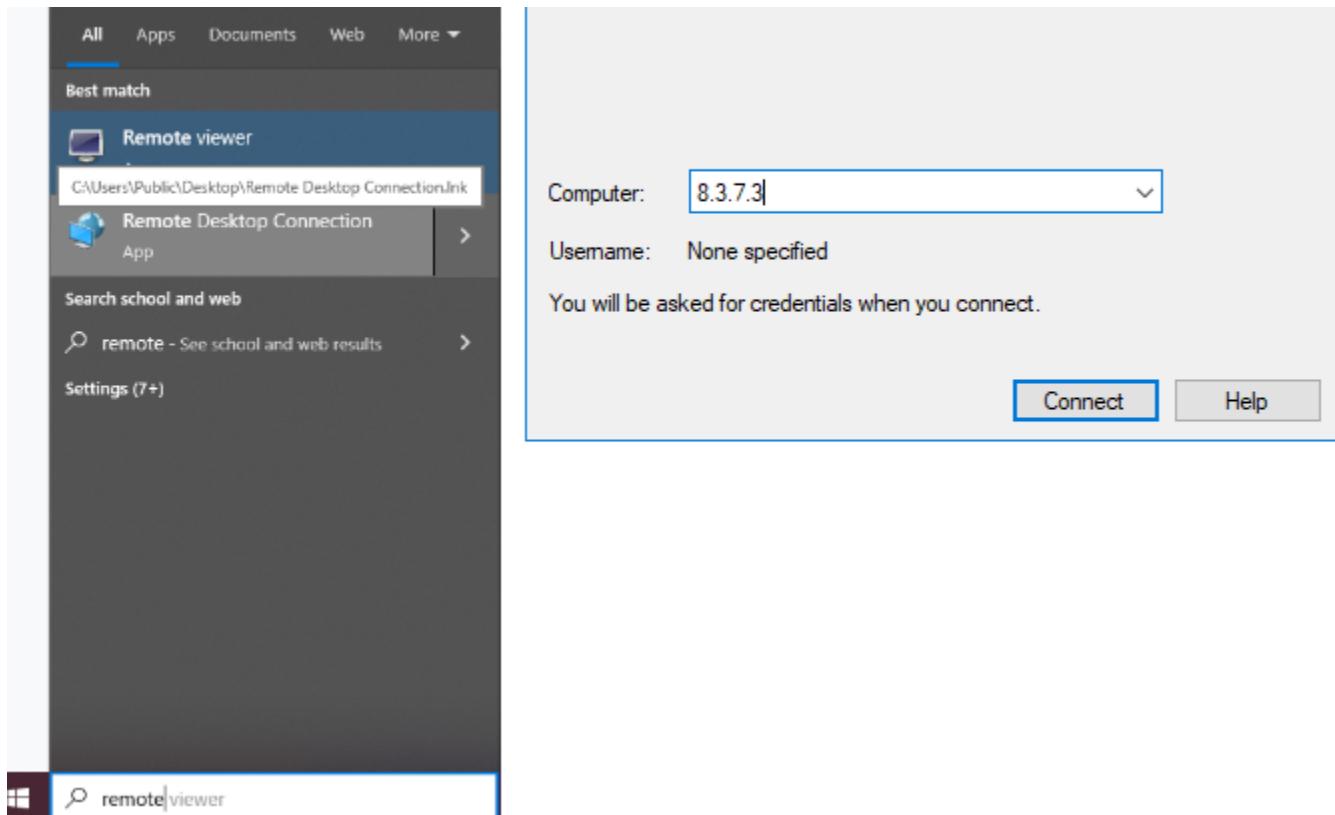
If you get the connection error.



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'.



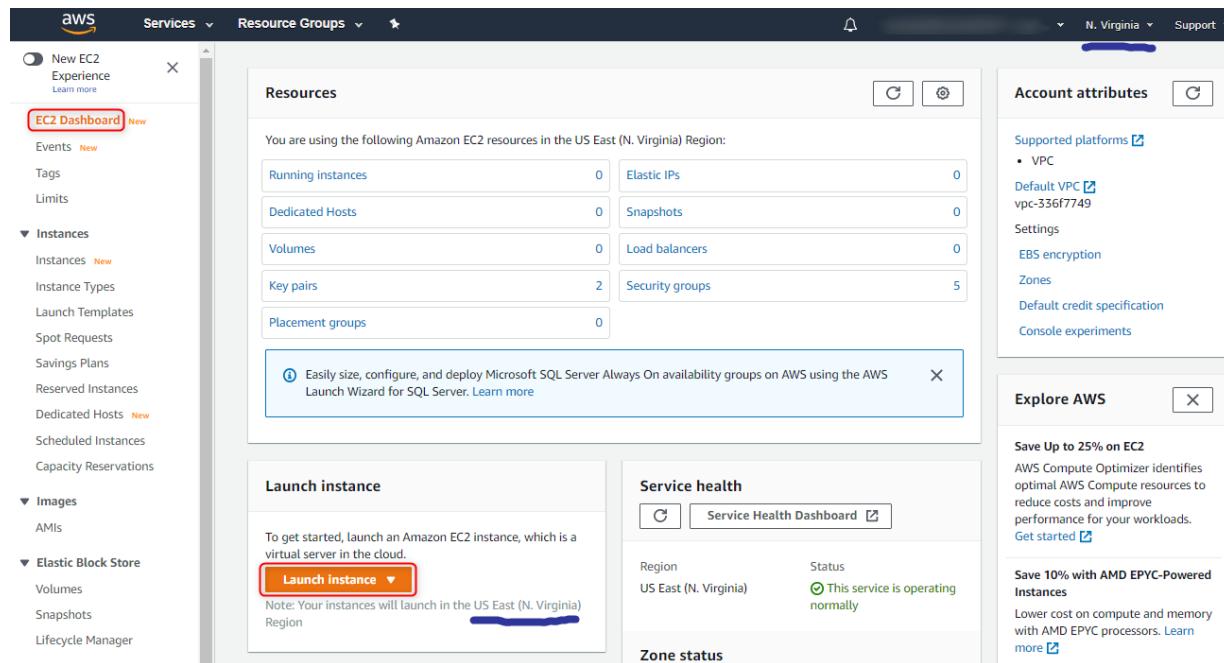
- 2) Select 'Another user.'
- 3) As the user name type 'Administrator'.
- 4) Decrypt Administrator's password from the .pem file as instructed above and use it to log in.

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.

- 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.

! If it isn't labeled "Free Tier Eligible", you may incur a charge!

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

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs
- Free tier only (i)

Image	Name	Description	Select	64-bit (x86)	64-bit (Arm)
Amazon Linux	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-02354e95b39ca8dec (64-bit x86)	Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.	Select	<input checked="" type="radio"/> 64-bit (x86)	<input type="radio"/> 64-bit (Arm)
Amazon Linux	Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0761dd91277e34178	The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.	Select	64-bit (x86)	

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

Step 2: Choose an Instance Type

Currently selected: t2.micro (Variable ECUs, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- 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):



'User data' is a method for bootstrapping your instance
- Any code placed here will be executed the first time an instance is launched.

```
#!/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://gitlab.com/karpenkoKhAI/ec2-template/-/raw/main/bootcamp-app.tar
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>.

The screenshot shows the AWS CloudFormation console interface. At the top, there's a navigation bar with tabs: Services, Resource Groups, N. Virginia, and Support. Below the navigation bar, there are seven steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance (which is underlined), 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review.

Step 3: Configure Instance Details

Advanced Details

User data

As text As file Input is already base64 encoded

```
#!/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
```

Buttons at the bottom:

- Cancel
- Previous
- Review and Launch** (highlighted with a red box)
- Next: Add Storage

5. Click **Next: Add Storage**:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-00e66c293c6f296fa	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Cancel Previous Review and Launch Next: Add Tags

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”.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	Linux Web Server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

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

6. 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.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:

- Create a new security group
- Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom	0.0.0.0/0 e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0 e.g. SSH for Admin Desktop

Add Rule

Cancel **Previous** **Review and Launch**

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

Note the warning box(es) 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.

Step 7: Review Instance Launch

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security Group ID	Name	Description
sg-0479f4917714d2594	Linux Web Server	Open ports SSH (22) and HTTP (80)

All selected security groups inbound rules

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
SSH	TCP	22	0.0.0.0/0	

Cancel **Previous** **Launch**

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

Select an existing key pair or create a new key pair

A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair
Select a key pair
lbuccd

I acknowledge that I have access to the selected private key file (lbuccd.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)

9. Now click the **Launch Instances** button to launch your linux web server.
10. 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.

The screenshot shows the AWS EC2 Launch Status page. At the top, there's a navigation bar with 'Services', 'Resource Groups', a star icon, a bell icon, 'Ohio', and 'Support'. Below the navigation is a section titled 'Launch Status' with a sub-section 'Get notified of estimated charges'. It says 'Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier)'. Under 'How to connect to your instances', it says 'Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.' It also says 'Click [View Instances](#) to monitor your instances' status. Below this, a 'Helpful resources' section lists links to the Amazon EC2 User Guide, Microsoft Windows Guide, and Discussion Forum. A note says 'While your instances are launching you can also' followed by links to Create status check alarms, Create and attach additional EBS volumes, and Manage security groups. A red box highlights the 'View Instances' button at the bottom right.

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.

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

	Name	Instance ID	Instance state	Instance type	Status check	Alarm St...
<input checked="" type="checkbox"/>	Linux Web Server	i-0841c39dcd9c69c18	Running	t2.micro	Initializing	No alarms +

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.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like New EC2 Experience, EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area displays a table of instances with the following columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm St..., and Availability zone. One instance is selected, showing details in a modal below:

Instances (1/2)						
	Name	Instance ID	Instance state	Instance type	Status check	Alarm St...
<input checked="" type="checkbox"/>	Linux Web Server	i-0841c39dcd9c69c18	Running	t2.micro	2/2 checks ...	No alarms +

Instance: i-0841c39dcd9c69c18 (Linux Web Server)

Details			Security	Networking	Storage	Status Checks	Monitoring	Tags
Instance summary Info								
Instance ID	Public IPv4 address		Private IPv4 addresses					
i-0841c39dcd9c69c18 (Linux Web Server)	54.80.153.230 open address		172.31.34.15					
Instance state	Public IPv4 DNS		Private IPv4 DNS					
Running	ec2-54-80-153-230.compute-1.amazonaws.com open address		ip-172-31-34-15.ec2.internal					
Instance type	VPC ID		Subnet ID					
t2.micro	vpc-336f7749		subnet-9126efce					

4. 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):

The screenshot shows a web browser window with the following details:

- Address Bar:** Not secure | ec2-35-153-51-96.compu...
- Content:**
 - Welcome to the Cloud Computing Development Module!**
 - This is your first AWS EC2 Linux Web Server**
 - Amazon Web Services Logo**
 - Navigation Links:** LOAD TEST, RDS
 - Meta-Data Table:**

Meta-Data	Value
InstanceId	i-0a56fef69e95787ea
Availability Zone	us-east-1e
 - System Status:** Current CPU Load: 0%
 - Success Message:** Great Job: You have built your first Linux web server! (with an exclamation mark icon)

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 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.

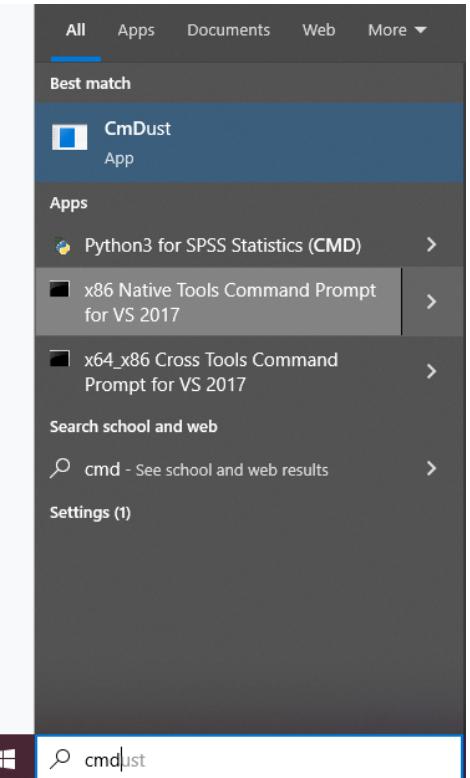
Make sure your Linux instance is running, select it and click on 'Connect'.

Instances (1/1) Info		C	Connect	Instance state ▾	Actions ▾	Launch instances	▼
<input type="text"/> Find instance by attribute or tag (case-sensitive)							
<input checked="" type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	⋮
<input checked="" type="checkbox"/>	MyLinux	i-07e11131737ca6e27	Running	t2.micro	2/2 checks passed	No alarms	+

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'



Connect to instance [Info](#)

Connect to your instance i-07e11131737ca6e27 (MyLinux) using any of these options

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

Instance ID
i-07e11131737ca6e27 (MyLinux)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is CCD.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 CCD.pem
4. Connect to your instance using its Public DNS:
ec2-44-211-204-217.compute-1.amazonaws.com

Example:
ssh -i "CCD.pem" ec2-user@ec2-44-211-204-217.compute-1.amazonaws.com

i **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.

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

Step 6: Terminate instances after use!!!



You may incur a charge if you limit your Free Tier!

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with navigation links: 'Lambda', 'Functions', 'Logs', 'Metrics', 'Actions', 'CloudWatch Metrics', 'CloudWatch Logs', and 'AWS CloudTrail'. The main area displays a table with one row. The row contains the Public IPv4 address '3.81.148.42', the instance ID 'Elast... (truncated)', and a dropdown menu with the value '-'. To the right of the table is a vertical 'Actions' menu. The menu items are: 'View details', 'Connect', 'Get Windows password', 'Create template from instance', 'Launch more like this', and 'Manage tags'. Below this is a section titled 'Stop instance' with options: 'Start instance', 'Reboot instance', 'Hibernate instance', 'Terminate instance', and 'Manage instance state'. The 'Terminate instance' option is highlighted in red. A secondary 'Actions' menu is open over this section, containing: 'Instance state' (with a right-pointing arrow), 'Instance settings' (with a right-pointing arrow), 'Networking' (with a right-pointing arrow), 'Image' (with a right-pointing arrow), and 'Monitoring' (with a right-pointing arrow).

References

IaaS PaaS on Amazon AWS EC2

<https://www.youtube.com/watch?v=H6QEGQBIWXs&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