

# **Cloud Server Project & Video Explainer**

## ***ICT171 Assignment 2***

**Student ID:** 35071098

**Student Name:** Neel Kamlesh Lachwani

## Table of Contents

GitHub Repository Link .....	3
Server Link.....	3
Introduction.....	5
Instance Server Launch.....	6
SSH Connect client .....	7

## List of Figures

Figure 1: Instance configurations.....	6
Figure 2: Key pair generation .....	6
Figure 3: Instance launched .....	6
Figure 4: Instance summary.....	7
Figure 5: SSH Connect client option .....	7
Figure 6: Connected to SSH Server .....	7
Figure 7: Download Apache service on EC2 Server .....	8
Figure 8: Installed .....	8
Figure 9: Start and enable the Apache Server .....	9
Figure 10: Replace the var index.html file.....	9
Figure 11: Blog Hosted website.....	9
Figure 12: Open Route 53 for domain register .....	10
Figure 13: Domain configurations .....	10

## GitHub Repository Link

<https://github.com/NEELLACHWANI/ICT171>

## Server Link

<http://3.27.234.91/>

*Table 1: Script Commands*

No.	Command	Description
1	ssh -i "connectKey.pem" ec2-user@ec2-3-27-234-91.ap-southeast-2.compute.amazonaws.com	Connects to the EC2 instance using SSH and a private key.
2	chmod 400 connectKey.pem	Sets secure permissions for the SSH key file.
3	sudo yum update -y	Updates all packages on the EC2 instance.
4	sudo yum install httpd -y	Installs the Apache HTTP server.
5	sudo systemctl start httpd	Starts the Apache web server.
6	sudo systemctl enable httpd	Enables Apache to start at boot.
7	sudo systemctl status httpd	Checks Apache's current status (running or not).
8	cd /var/www/html	Navigates to the web root directory.
9	ls -la	Lists all files in long format, including hidden ones.
10	sudo nano index.html	Opens the index file in the Nano text editor for editing.
11	sudo rm index.html	Deletes the existing index.html file.
12	sudo cp /home/ec2-user/yourfile.html /var/www/html/index.html	Copies your custom HTML file to the hosting directory.

<b>13</b>	<code>sudo chown -R ec2-user:ec2-user /var/www/html</code>	Changes ownership so ec2-user can modify files.
<b>14</b>	<code>git clone</code>	Clones code from a GitHub repo into the EC2 instance.
<b>15</b>	<code>sudo reboot</code>	Reboots the instance (use with caution).

## **Introduction**

In this project, the deployment of a fully functional blog site is displayed on an Amazon Web Services (AWS) EC2 instance. This was done by spawning a virtual server, securely logging in through SSH and then installing the Apache HTTP web server to serve web content. The blog was hosted by deploying a custom blog written in HTML, CSS, and JavaScript by setting the root directory of the server. Version control and source code management were done on GitHub, whereas AWS Route 53 was utilized in domain registration. This project demonstrates useful skills in cloud computing, setting up servers and web hosting that offers a cheap and scalable solution to hosting personal or professional websites.

# Instance Server Launch

**Launch an instance** [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [Info](#)

Name  
ICT171-NeelServer [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents [Quick Start](#)

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Summary**

Number of instances [Info](#)  
1

**Software image (AMI)**  
Amazon Linux 2023 AMI 2023.7.2...[read more](#)  
ami-06a0b33485e9d1cf1

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
New security group

**Storage (volumes)**  
1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Preview code](#)

Figure 1: Instance configurations

**Create key pair**

**Key pair name**  
Key pairs allow you to connect to your instance securely.  
connectKey  
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

☒ RSA  
RSA encrypted private and public key pair

☐ ED25519  
ED25519 encrypted private and public key pair

**Private key file format**

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

[Cancel](#) [Create key pair](#)

Figure 2: Key pair generation

**Success**  
Successfully initiated launch of instance (i-0dd2949dd05d5e9be)

**Launch log**

Initializing requests	✓ Succeeded
Creating security groups	✓ Succeeded
Creating security group rules	✓ Succeeded
Launch initiation	✓ Succeeded

**Next Steps**

What would you like to do next with this instance, for example "create alarm" or "create backup"

< 1 2 3 4 5 6 >

Figure 3: Instance launched

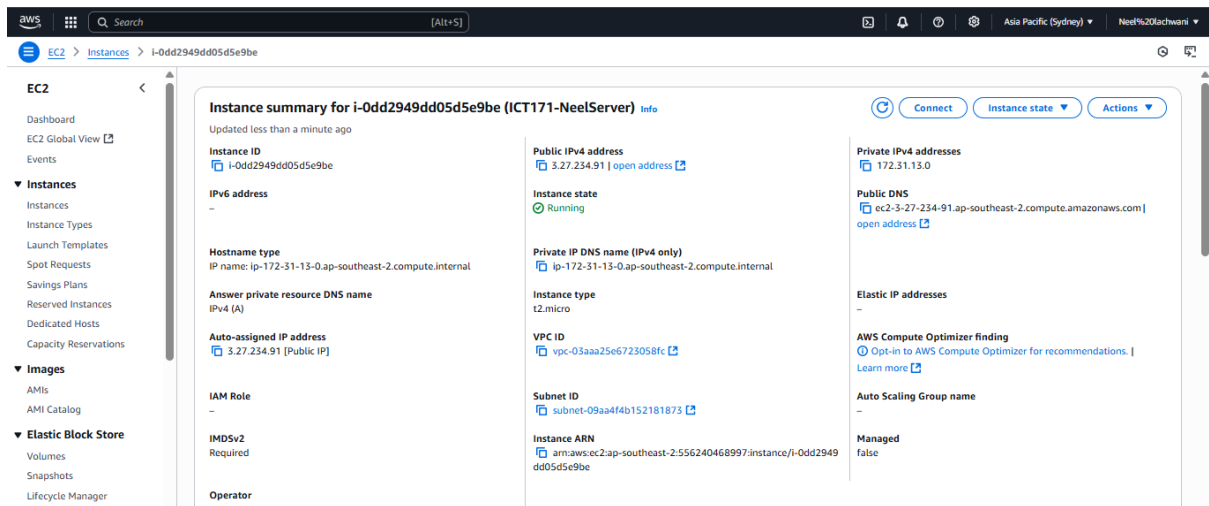


Figure 4: Instance summary

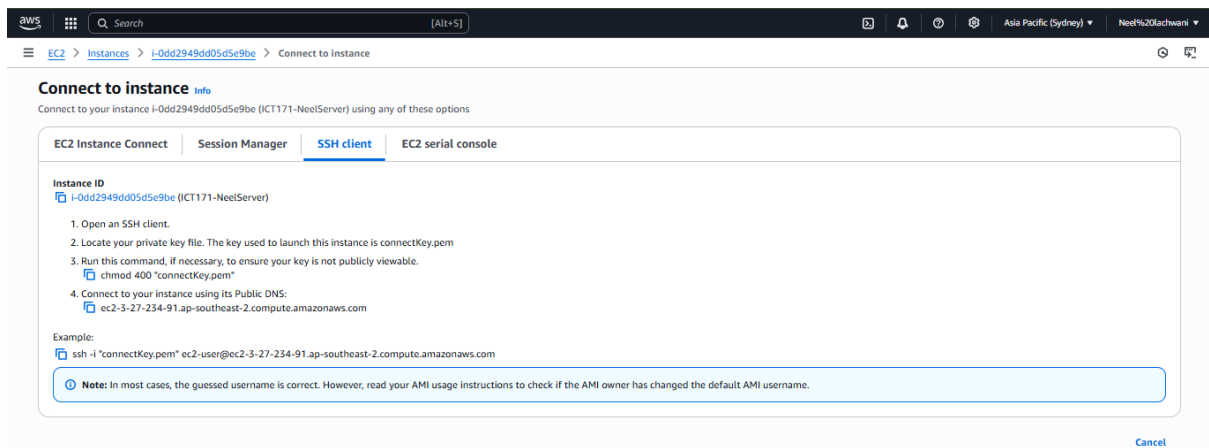


Figure 5: SSH Connect client option

## SSH Connect client

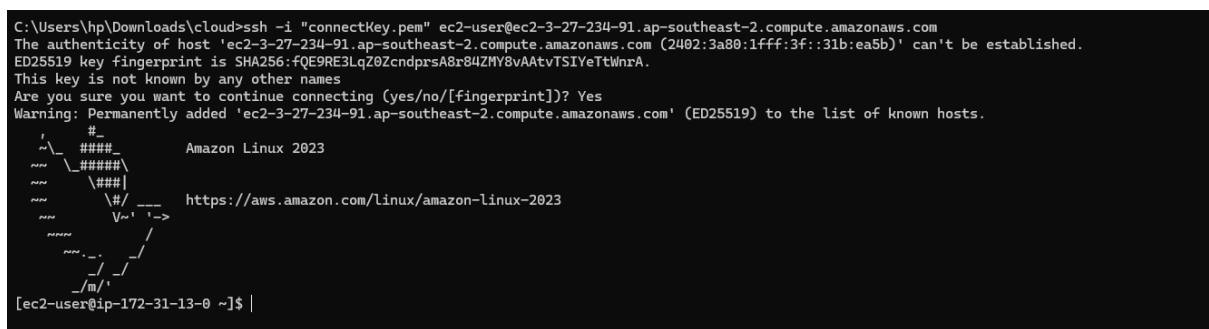


Figure 6: Connected to SSH Server

## Command:

“ssh -i "connectKey.pem" [ec2-user@ec2-3-27-234-91.ap-southeast-2.compute.amazonaws.com](https://ec2-3-27-234-91.ap-southeast-2.compute.amazonaws.com/)”

## Description:

It allows you to safely access an AWS EC2 instance via this SSH command. The private key file is given to `-i` in this flag such as `connectKey.pem`. By default, a user named `ec2-user` will be created on Amazon Linux. The hostname is a short form that takes you to the public DNS of the target EC2 in the Asia Pacific (Sydney) region. Using this method, communication is protected using keys, eliminating the need for passwords.

```
ec2-user@ip-172-31-13-0:~$ sudo yum install httpd -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
156 kB/s | 16 kB    00:00
```

Package	Architecture	Version	Repository	Size
Installing:				
httpd	x86_64	2.4.62-1.amzn2023	amazonlinux	48 k
Installing dependencies:				
apr	x86_64	1.7.5-1.amzn2023.0.4	amazonlinux	129 k
apr-util	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	98 k
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
httpd-core	x86_64	2.4.62-1.amzn2023	amazonlinux	1.4 M
httpd-filesystem	noarch	2.4.62-1.amzn2023	amazonlinux	14 k
httpd-tools	x86_64	2.4.62-1.amzn2023	amazonlinux	81 k
libbrotli	x86_64	1.0.9-4.amzn2023.0.2	amazonlinux	315 k
mailcap	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	33 k
Installing weak dependencies:				
apr-util-openssl	x86_64	1.6.3-1.amzn2023.0.1	amazonlinux	17 k
mod_http2	x86_64	2.0.27-1.amzn2023.0.3	amazonlinux	166 k
mod_lua	x86_64	2.4.62-1.amzn2023	amazonlinux	61 k

```
Transaction Summary
Install 12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm           314 kB/s | 17 kB    00:00
(2/12): apr-1.7.5-1.amzn2023.0.4.x86_64.rpm                     2.1 MB/s | 129 kB   00:00
(3/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm                 1.4 MB/s | 98 kB    00:00
(4/12): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm    881 kB/s | 19 kB    00:00
(5/12): httpd-2.4.62-1.amzn2023.x86_64.rpm                       2.2 MB/s | 48 kB    00:00
(6/12): httpd-filesystem-2.4.62-1.amzn2023.noarch.rpm            510 kB/s | 14 kB    00:00
```

Figure 7: Download Apache service on EC2 Server

```
ec2-user@ip-172-31-13-0:~$ sudo yum install httpd -y
Installing      : apr-1.7.5-1.amzn2023.0.4.x86_64                1/12
Installing      : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 2/12
Installing      : apr-util-1.6.3-1.amzn2023.0.1.x86_64         3/12
Installing      : mailcap-2.1.49-3.amzn2023.0.3.noarch          4/12
Installing      : httpd-tools-2.4.62-1.amzn2023.x86_64         5/12
Installing      : libbrotli-1.0.9-4.amzn2023.0.2.x86_64        6/12
Running scriptlet: httpd-filesystem-2.4.62-1.amzn2023.noarch    7/12
Installing      : httpd-filesystem-2.4.62-1.amzn2023.noarch    7/12
Installing      : httpd-core-2.4.62-1.amzn2023.x86_64         8/12
Installing      : mod_http2-2.0.27-1.amzn2023.0.3.x86_64       9/12
Installing      : mod_lua-2.4.62-1.amzn2023.x86_64            10/12
Installing      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 11/12
Installing      : httpd-2.4.62-1.amzn2023.x86_64              12/12
Running scriptlet: httpd-2.4.62-1.amzn2023.x86_64              12/12
Verifying      : apr-1.7.5-1.amzn2023.0.4.x86_64                1/12
Verifying      : apr-util-1.6.3-1.amzn2023.0.1.x86_64         2/12
Verifying      : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/12
Verifying      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 4/12
Verifying      : httpd-2.4.62-1.amzn2023.x86_64               5/12
Verifying      : httpd-core-2.4.62-1.amzn2023.x86_64         6/12
Verifying      : httpd-filesystem-2.4.62-1.amzn2023.noarch    7/12
Verifying      : httpd-tools-2.4.62-1.amzn2023.x86_64       8/12
Verifying      : libbrotli-1.0.9-4.amzn2023.0.2.x86_64       9/12
Verifying      : mailcap-2.1.49-3.amzn2023.0.3.noarch        10/12
Verifying      : mod_http2-2.0.27-1.amzn2023.0.3.x86_64     11/12
Verifying      : mod_lua-2.4.62-1.amzn2023.x86_64           12/12

Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64    apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.62-1.amzn2023.x86_64          httpd-core-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch  httpd-tools-2.4.62-1.amzn2023.x86_64    libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch      mod_http2-2.0.27-1.amzn2023.0.3.x86_64  mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[ec2-user@ip-172-31-13-0 ~]$
```

Figure 8: Installed

## Command:

“`sudo yum install httpd -y`”



This installs the Apache HTTP server on Amazon Linux or CentOS using the command `sudo yum install httpd -y`. The `sudo` prefix is needed for installing software because it needs permission from the superuser. That command installs the Apache web server, so the system can serve website content. The `-y` flag makes sure that any prompts are automatically approved, so the entire installation can take place without needing user input.

Figure 9: Start and enable the Apache Server

Figure 10: Replace the var index.html file



9

This is due to the fact that the effective presentation of the site in the browser window is the confirmation that the implementation of a web server on the Amazon EC2 instance has been successfully carried out. It was done by spinning a virtual machine on Amazon EC2 and then securely connecting to it over SSH with a private key.

“sudo yum install httpd -y”

“sudo systemctl start httpd

sudo systemctl enable httpd”

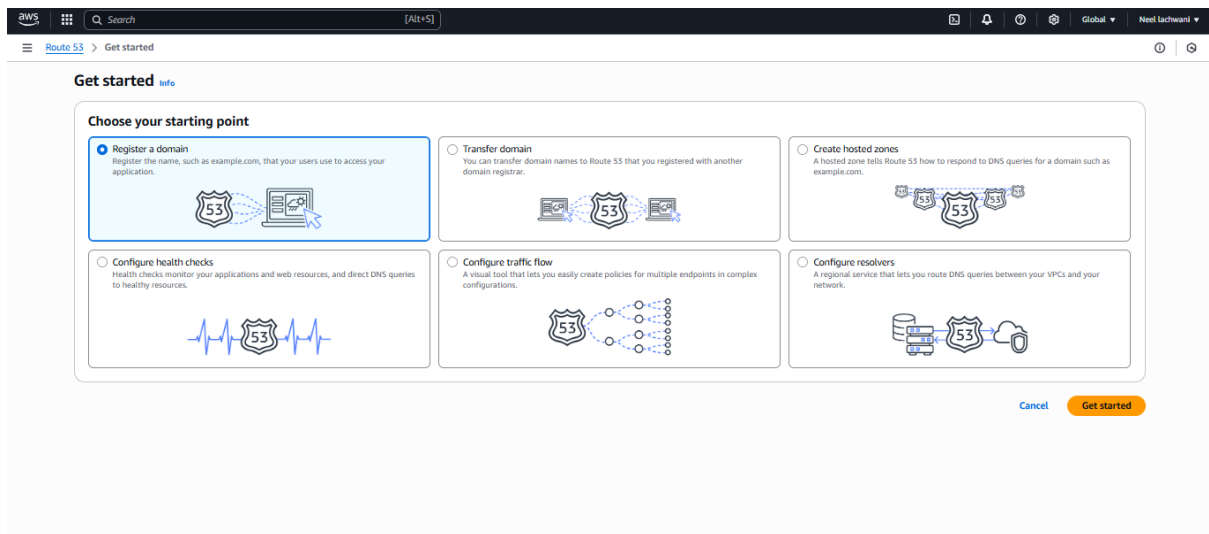


Figure 12: Open Route 53 for domain register

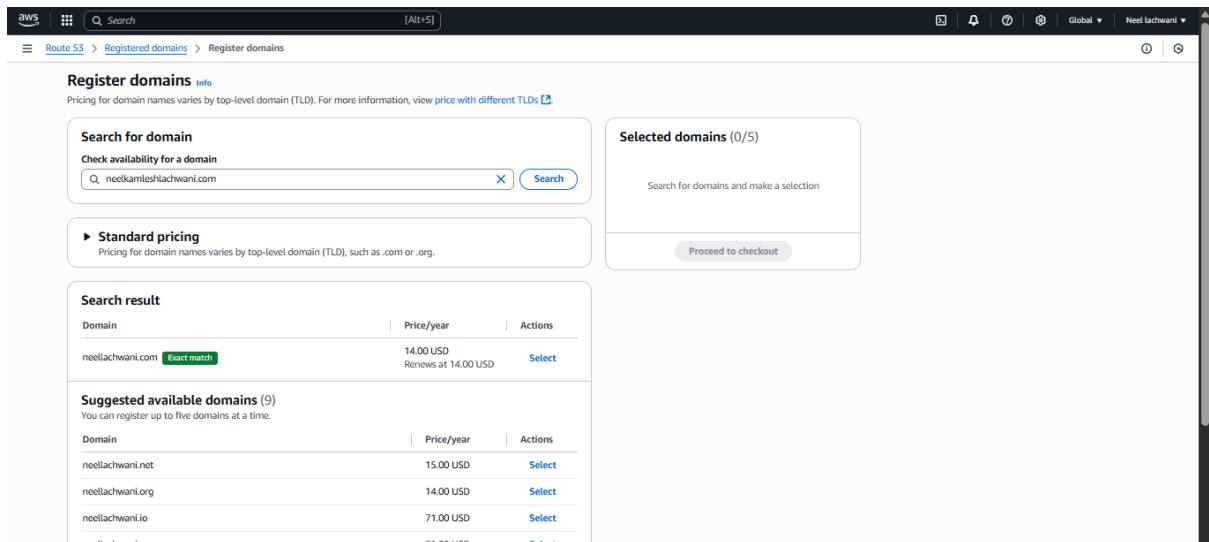


Figure 13: Domain configurations

The AWS Route 53 domain registration page in which a custom domain is searched and selected. DNS (Domain Name System) is a service that converts this human-readable domain to an IP address to access the web. TLS (Transport Layer Security) is the technology that

provides secure and encrypted communication between the server and the browser that a user uses. Professional websites should be registered and protected via TLS as it contributes to the ease of access as well as trust. The effect of this process is that the blog is hosted on a personalized secure and easy to remember address.