

Name - Priyanshu Lapkale

Class - TY B

Roll No - 322067

PRN - 22220008

Assignment 3

Deploye web application on AWS Cloud

Cloud Computing Definition

- Cloud computing refers to the use of hosted services, such as data storage, servers, databases, networking, and software over the internet. The data is stored on physical servers, which are maintained by a cloud service provider. Computer system resources, especially data storage and computing power, are available on-demand, without direct management by the user in cloud computing.

There are four cloud deployment models: public, private, community, and hybrid. Each deployment model is defined according to where the infrastructure for the environment is located. There are three main cloud service models: Software as a Service, Platform as a Service, and Infrastructure as a Service.

First you have to create a free account on AWS to get started. For this it's mandatory to enter details of a Debit/Credit card. Deploy Web application on AWS Cloud (or any cloud)(PHP/Python/Node js any application)

Create an EC2 instance

- Go to services and search for EC2 and click on it
- Then click on Launch intance

1. Set a name for your EC2 instance

This name should be unique and should be understandable for what purpose we made this instance.

Name and tags [Info](#)

Name

[Add additional tags](#)

2. Choose AMI(Amazon Machine Type)

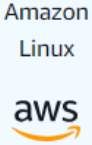
Technically we are using amazon's hardware so we have to define which operating system we want on our virtual computer. There are multiple options available but we'll be going with AMAZON Linux 2 AMI.

We'll use t2.micro because we only want to host a single static page.(Moreover because it's free XD)


▼ 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


Quick Start




Amazon Linux




macOS




Ubuntu




Windows



Red Hat



SUSE Linux



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-0449c34f967dbf18a (64-bit (x86), uefi-preferred) / ami-0796d19ad30229ab5 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2023 AMI 2023.3.20240205.2 x86_64 HVM kernel-6.1

Architecture

64-bit (x86) ▼

Boot mode

uefi-preferred

AMI ID

ami-0449c34f967dbf18a

Verified provider

3. Select the instance type Amazon provide a wide variety of instance based on our use case. They differ based on CPUs, Memory, Storage, etc.

Instance types (1/510+) Get advice							
<input type="text" value="Find resources by attribute or tag"/>							
Instance type	vCPUs	Architecture	Memory (GiB)	Storage (GB)	Storage type	Network performance	
<input type="radio"/> t2.nano	1	i386, x86_64	0.5	-	-	Low to Moderate	
<input checked="" type="radio"/> t2.micro	1	i386, x86_64	1	-	-	Low to Moderate	
<input type="radio"/> t2.small	1	i386, x86_64	2	-	-	Low to Moderate	
<input type="radio"/> t2.medium	2	i386, x86_64	4	-	-	Low to Moderate	
<input type="radio"/> t2.large	2	x86_64	8	-	-	Low to Moderate	

4. Set Key Pair(login)

- Click on create new key pair and set a unique key name e.g. icy-key-pair
- Select Key pair type (which security you want) e.g. RSA

- Select Private key file format (We'll need these key file later) e.g. .pem

Create key pair ✕

Key pair name
Key pairs allow you to connect to your instance securely.

Enter key pair name

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


Select your new key -

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

icy-key-pair ▼

 [Create new key pair](#)

5. Set Network Setting

In network setting we have to define rules and boundaries, like how your instance will react to the outer world or let's say internet. We'll allow SSH traffic so that we can connect with our instance. HTTP traffic

also because we are hosting a website and we want to access it via our browser.

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-039948e0d48e3f441

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-2' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere
0.0.0.0/0

☐ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

⚠ 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.

×

6. Configure storage

We can add more volume to our instance. But it'll depend on scale of our project and it'll also require money. We'll go with default i.e. 8 which is free tier.

▼ **Configure storage** [Info](#)

Advanced

1x GiB ▼ Root volume (Not encrypted)

📘 Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

✕

Add new volume

🕒 Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

🔄

0 x File systems [Edit](#)

Now Once Again Review the details and click on Launch Instance button.

Now Let's connect with our EC2 instance using SSH

1. Open your SSH client (Terminal on your machine) and then go to the location where your .pem file is saved. (Refer above Step 4 in Creating an instance)
2. Run this command, if necessary, to ensure your key is not publicly viewable.

```
$ chmod 400 [your-key-name].pem
```

3. In AWS console select your instance and click on the connect button on the top

Instances (1/1) Info										
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>				Any state ▼						
☑	Name ↗	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IPv4 DNS ▼	Public IPv4 ... ▼	Elastic IP
☑	lcy-test-server	i-016b818223317b78f	🟢 Running 🔍	t2.micro	🕒 Initializing	View alarms +	ap-south-1a	ec2-52-66-209-114.ap-...	52.66.209.114	-

4. We want to connect through SSH Client, so click on it and copy paste the given command in your terminal.


EC2 Instance Connect



Session Manager

SSH client


EC2 serial console


Instance ID

 **i-016b818223317b78f** (icy-test-server)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is icy-key-pair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 `chmod 400 "icy-key-pair.pem"`
4. Connect to your instance using its Public DNS:
 `ec2-52-66-209-114.ap-south-1.compute.amazonaws.com`

Example:

 `ssh -i "icy-key-pair.pem" ec2-user@ec2-52-66-209-114.ap-south-1.compute.amazonaws.com`

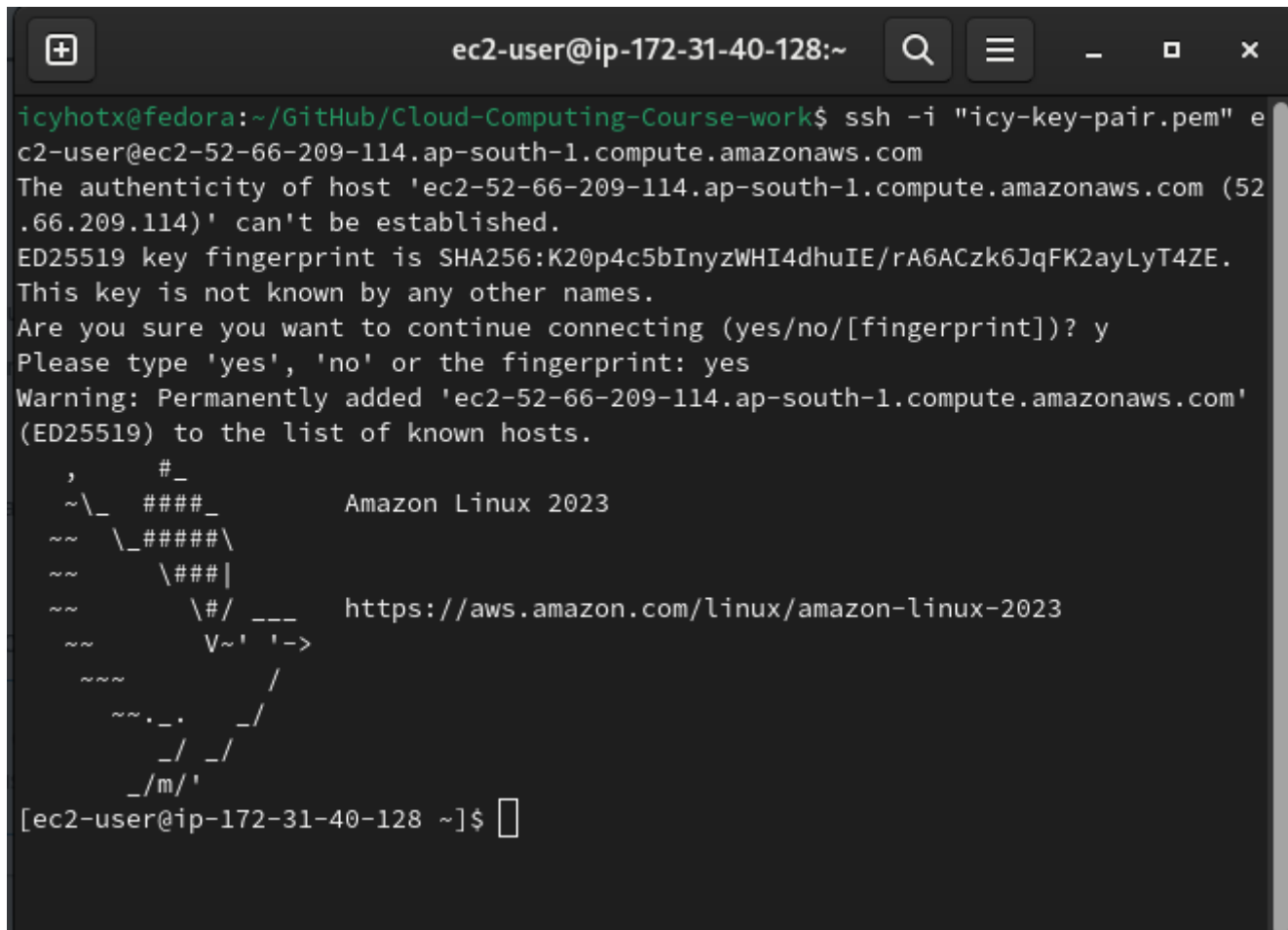
 **Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

In my case my command is this -

```
$ ssh -i "icy-key-pair.pem" ec2-user@ec2-52-66-209-114.ap-south-1.compute.amazonaws.com
```

To break it down -

ssh command is used to provide secure encrypted connection between two host. Then we have one option **-i** which says install, then there is our key-pair name followed by user along with the public DNS of our server.

A terminal window titled 'ec2-user@ip-172-31-40-128:~' showing an SSH session. The user 'icyhotx' at 'fedora' connects to 'ec2-52-66-209-114.ap-south-1.compute.amazonaws.com'. The terminal displays a warning about the host's authenticity, the ED25519 key fingerprint, and a confirmation to add the host to the known hosts list. After connecting, the user is greeted with the Amazon Linux 2023 logo and a URL to the AWS Linux documentation. The prompt is '[ec2-user@ip-172-31-40-128 ~]\$' with a cursor.

```
ec2-user@ip-172-31-40-128:~
icyhotx@fedora:~/GitHub/Cloud-Computing-Course-work$ ssh -i "icy-key-pair.pem" e
c2-user@ec2-52-66-209-114.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-52-66-209-114.ap-south-1.compute.amazonaws.com (52
.66.209.114)' can't be established.
ED25519 key fingerprint is SHA256:K20p4c5bInyzWHI4dhuIE/rA6ACzk6JqFK2ayLyT4ZE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'ec2-52-66-209-114.ap-south-1.compute.amazonaws.com'
(ED25519) to the list of known hosts.

,      #_
~\_   #####_      Amazon Linux 2023
~~  \_#####\
~~   \###|
~~   \#/  ___  https://aws.amazon.com/linux/amazon-linux-2023
~~      V~'  '->
~~~~
~~.  _  _/_/
  _/_/_/
    _/m/'
[ec2-user@ip-172-31-40-128 ~]$
```

Installing a Web Server

1. Elevate your privileges

```
$ sudo su
```

2. Update all packages

```
# yum update -y
```

3. Installing the apache webserver

```
# yum install httpd -y
```

4. Start the webserver

```
# service httpd start
```

5. Add an index page in your server

```
# cd /var/www/html
html# nano index.html
```

- 6. Enter your index.html code
- For just testing purpose add only one sentence

Type

```
This is icyy, and this is our first EC2 server that we hosted.
```

Press ctrl+x , then press y and Enter.

Checking Result

Now go back to EC2 instance page and select our instance go to details section and open the Public IPv4 address in another tab -

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive) Any state

Connect

Instance state

Actions

Launch instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elas
Icy-test-webse...	i-06e750eb8b28164bb	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-43-204-215-146.ap...	43.204.215.146	-

Instance: i-06e750eb8b28164bb (icy-test-webserver)

Details Status and alarms New Monitoring Security Networking Storage Tags

Instance summary Info

Instance ID

i-06e750eb8b28164bb (icy-test-webserver)

Public IPv4 address

43.204.215.146 [open address](#)

Private IPv4 addresses

172.31.11.191

Not secure

43.204.215.146

This is icyy, and this is our first EC2 server that we hosted.

Here we can conclude that we've hosted a static webpage using EC2 in AWS.We can also add more html pages to make it more interactive.