

3. Deploy Web application on AWS Cloud (or any cloud)

- Name: Om Deore
- PRN: 22110318
- Div: TY(A)
- RollNo: 321022

1. What is cloud computing definition

Cloud computing is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. Large clouds often have functions distributed over multiple locations, each of which is a data center. Cloud computing relies on sharing of resources to achieve coherence and typically uses a pay-as-you-go model, which can help in reducing capital expenses but may also lead to unexpected operating expenses for users.

- Wikipedia

2. Cloud service and deployment models

IaaS, PaaS and SaaS are the three most popular types of **cloud service** offerings. They are sometimes referred to as cloud service models or cloud computing service models.

1. IaaS, or infrastructure as a service, is on-demand access to cloud-hosted physical and virtual servers, storage and networking—the backend IT infrastructure for running applications and workloads in the cloud.
2. PaaS, or platform as a service, is on-demand access to a complete, ready-to-use, cloud-hosted platform for developing, running, maintaining and managing applications.
3. SaaS, or software as a service, is on-demand access to ready-to-use, cloud-hosted application software.

The **cloud deployment** model identifies the specific type of cloud environment based on ownership, scale, and access, as well as the cloud's nature and purpose.

1) Public Cloud

Anybody can access systems and services, infrastructure is owned by the entity that delivers the cloud services

2) Private Cloud

one-on-one environment for a single user (customer). There is no need to share your hardware

with anyone else. The distinction between private and public clouds is in how you handle all of the hardware

The advantages of using a private cloud are as follows:

3) Hybrid Cloud

By bridging the public and private worlds with a layer of proprietary software, hybrid cloud computing gives the best of both worlds. With a hybrid solution, you may host the app in a safe environment while taking advantage of the public cloud's cost savings

4) Community Cloud

It allows systems and services to be accessible by a group of organizations. It is a distributed system that is created by integrating the services of different clouds to address the specific needs of a community, industry, or business.

5) Multi-Cloud

We're talking about employing multiple cloud providers at the same time under this paradigm, as the name implies. It's similar to the hybrid cloud deployment approach, which combines public and private cloud resources.

3) Step-by-step screenshot to upload web application on the cloud

1. Instance

Step 1: Login to [AWS console](#)

Step 2: go to ec2 and click on launch instance

Step 3: Give your instance a name and tags if required, choose what operating system you want on your AWS server. Select CPU type

Here we choose ubuntu 22.04.4 LTS as our OS and t3.micro for CPU type

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. The current step is 'Name and tags'. The user has entered 'CC-Assignment' as the instance name and selected the 'Add additional tags' option. Below this, the 'Application and OS Images (Amazon Machine Image)' section is shown, featuring a search bar and a catalog entry for 'ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20240301'. The instance is marked as a 'Verified provider' and 'Free tier eligible'. The 'Instance type' section shows the 't3.micro' instance type selected, with details about its performance and pricing. The 'Summary' section on the right provides an overview of the launch configuration, including the number of instances (1), software image (Ubuntu Server 22.04 LTS (HVM)), virtual server type (t3.micro), and storage (1 volume - 8 GB). A callout box highlights the 'Free tier' benefits. The final step is 'Review commands' before launching.

Step 4: Create a key pair to securely connect to your instance.

Step 5: Configure network security groups, storage and launch instance

The screenshot shows the AWS EC2 Launch Instances wizard at Step 5. The top navigation bar includes the AWS logo, a search bar, and the URL <https://eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LaunchInstances>.

Key pair name - required: cc_aws_key (dropdown menu) | Create new key pair [Alt+S]

Network settings (Info) | Edit

Network | Info
vpc-036d9116287f7cf37

Subnet | Info
No preference (Default subnet in any availability zone)

Auto-assign public IP | Info
Enable

Additional charges apply when outside of free tier allowance

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-1' with the following rules:

Allow SSH traffic from Anywhere 0.0.0.0/0
Helps you connect to your instance

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

Configure storage (Info) | Advanced

1x 8 GiB gp2 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

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

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

Summary

Number of instances | Info
1

Software Image (AMI)
Ubuntu Server 22.04 LTS (HVM),...read more
ami-0914547665e6a707c

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch Instance | Review commands

Once launched you can see it running in instances tab

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area is titled 'Instances (1) Info' and shows a table with one row. The row details an instance named 'CC-Assignment' with Instance ID 'i-058d3efc0d0d1d19c'. The instance is 'Running' (status check 'OK'), type 't3.micro', and in 'eu-north-1b'. A search bar at the top says 'Find Instance by attribute or tag (case-sensitive)' and a dropdown says 'All states'. Below the table, a section titled 'Select an instance' is visible.

Step 6: ssh into our AWS instance

To ssh into our instance we need to have our key at ready on local machine. Change permissions of that key.pem file to 400 so only current user can access it in read-only mode

```
$ chmod 400 key.pem
```

To connect to your server use following command

```
$ ssh -i "cc_aws_key.pem" ubuntu@aws_server_dns.compute.amazonaws.com
```

this takes you into aws server's shell and prompt

● ● ● ●

~ : bash – Konsole

08 Apr 2024 17:23 [10/31]

set to 400 permissions te.amazonaws.com

Local Computer's Shell

```
om_d@deblinu:~/Codes/server
> nvim ~/.config/nvim/init.lua
```

AWS Server's Shell (accessed via ssh)

```
om_d@deblinu:~/Codes/server
> neofetch
      _met$$$$$gg.
     ,g$$$$$$$$$$$$$$$$$P. -----
     ,g$$P"    """Y$$. .
     ,$$P'           '$$$. OS: Debian GNU/Linux 12 (bookworm) x86_64
     '$$P      ,ggs.   '$$b: Host: Inspiron 14 5418
     `d$$'   ,$P'" .   $$$ Uptime: 1 hour, 3 mins
     $$P   d$'   ,   $$P Packages: 3263 (dpkg)
     $$:   $$-.   - ,d$$' Shell: bash 5.2.15
     $$;   Y$b._   ,d$P' Resolution: 1920x1080
     Y$$.   . "Y$$$P"
     '$$b"  "-___ WM: kwin
     `Y$$
     `Y$$. WM Theme: Mkos-BigSur-Night
     `$$b. Theme: [Plasma], Ant-Dark [GTK2/3]
     `Y$$b.
     `Y$b.. Icons: kora-light-panel [Plasma], kora-light-panel [GTK2/3]
     `Y$b._ CPU: 11th Gen Intel i5-11320H (8) @ 4.500GHz
```

No VM guests are running outdated hypervisor (qemu) binaries on this host.

```
ubuntu@ip-172-31-42-235:~$ neofetch
      .-/+oossssoo/+-.
      :+ssssssssssssssssssss+:`----- ubuntu@ip-172-31-42-235
      +ssssssssssssssssssssyssss+- OS: Ubuntu 22.04.4 LTS x86_64
      .osssssssssssssssssssdMMMyssso. Host: t3.micro
      /sssssssssshdmmNNmyNMMMHssssss/ Kernel: 6.5.0-1014-aws
      +ssssssssshydMMMMMMNdddyssssssss+ Uptime: 15 mins
      /ssssssssshhyyyyhmNMMMNhssssssss/ Packages: 692 (dpkg), 6 (snap)
      .ssssssssdMMMNhsssssssssshNMMMdssssss. Shell: bash 5.1.16
      +sssshhhyNMMNyssssssssssyNMMMyssssss+ Terminal: /dev/pts/0
      ossyNMMNyMhssssssssssssssshmmhssssssso CPU: Intel Xeon Platinum 8259CL (2) @ 2.499GHz
      ossyNMMNyMhssssssssssssssshmmhssssssso GPU: 00:03.0 Amazon.com, Inc. Device 1111
      +sssshhhyNMMNyssssssssssyNMMMyssssss+ Memory: 172MiB / 904MiB
      .ssssssssdMMMNhsssssssssshNMMMdssssss.
      /sssssssshNMMNyhyyyyhdNMMMNhssssssss/
      +sssssssssdmydMMMMMMNdddyssssssss+-----[5/1181]
      /sssssssssshdmNNNmyNMMMHssssss/
      .osssssssssssssssssdMMMNyssso.
      +ssssssssssssssssssyssss+-
```

"deblinu" 17:23 08-Apr-24

Step 7: Edit network security group permissions to allow traffic on http protocol's port

The screenshot shows the AWS EC2 Network Security Group inbound rules configuration. It lists two rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0308c7ca1c58566c1	SSH	TCP	22	Custom	0.0.0.0/0
-	HTTP	TCP	80	Anywhere... 0.0.0.0/0	0.0.0.0/0

A yellow warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." Buttons for "Cancel", "Preview changes", and "Save rules" are at the bottom right.

2. LAMP installation

LAMP stands for

1. L - Linux for Operating system
2. A - Apache HTTP server
3. M - mysql for relational database management system
4. P - php, perl or python as programming language

We installed L in our linux server, let's install other three on this server too

step 1: Install rest of packages

```
$ sudo apt-get install apache2
$ sudo apt-get install mysql-server
$ sudo apt-get install php php-mysql
```

step 2: Change permissions of /var/www/html/ folder

```
$ sudo chmod 777 -R /var/www/html
```

3. MySQL

step 3: Create a user in MySQL and give that user all database privileges

This screenshot shows a terminal session on an Ubuntu 22.04 LTS server. The user is setting up a MySQL database and installing the LAMP stack.

MySQL Setup:

- Line 1: `ubuntu@ip-172-31-42-235:~\$ sudo mysql`
- Line 2: Welcome to the MySQL monitor. Commands end with ; or \g.
- Line 3: Your MySQL connection id is 14
- Line 4: Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)
- Line 5: Copyright (c) 2000, 2024, Oracle and/or its affiliates.
- Line 6: Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
- Line 7: Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
- Line 8: mysql> CREATE USER 'user1'@'localhost' IDENTIFIED BY 'password';
- Line 9: ERROR 1396 (HY000): Operation CREATE USER failed for 'user1'@'localhost'
- Line 10: mysql> CREATE USER 'user2'@'localhost' IDENTIFIED BY 'password';
- Line 11: Query OK, 0 rows affected (0.01 sec)
- Line 12: mysql> GRANT ALL PRIVILEGES ON *.* TO 'user2'@'localhost' WITH GRANT OPTION;
- Line 13: Query OK, 0 rows affected (0.01 sec)
- Line 14: mysql> FLUSH PRIVILEGES;
- Line 15: Query OK, 0 rows affected (0.01 sec)
- Line 16: mysql>

LAMP Stack Installation:

- Line 1: `ubuntu@ip-172-31-42-235:~\$ bash`
- Line 2: `root@ip-172-31-42-235:~# apt-get update`
- Line 3: `root@ip-172-31-42-235:~# apt-get upgrade`
- Line 4: `root@ip-172-31-42-235:~# apt-get install apache2`
- Line 5: `root@ip-172-31-42-235:~# apt-get install mysql-server`
- Line 6: `root@ip-172-31-42-235:~# apt-get install php php-mysql`
- Line 7: `root@ip-172-31-42-235:~#`
- Line 8: `root@ip-172-31-42-235:~#`
- Line 9: `root@ip-172-31-42-235:~#`
- Line 10: `root@ip-172-31-42-235:~#`
- Line 11: `root@ip-172-31-42-235:~#`
- Line 12: `root@ip-172-31-42-235:~#`
- Line 13: `root@ip-172-31-42-235:~#`
- Line 14: `root@ip-172-31-42-235:~#`
- Line 15: `root@ip-172-31-42-235:~#`
- Line 16: `root@ip-172-31-42-235:~#`
- Line 17: `root@ip-172-31-42-235:~#`
- Line 18: `root@ip-172-31-42-235:~#`
- Line 19: `root@ip-172-31-42-235:~#`
- Line 20: `root@ip-172-31-42-235:~#`
- Line 21: `root@ip-172-31-42-235:~#`
- Line 22: `root@ip-172-31-42-235:~#`
- Line 23: `root@ip-172-31-42-235:~#`
- Line 24: `root@ip-172-31-42-235:~#`
- Line 25: `root@ip-172-31-42-235:~#`
- Line 26: `root@ip-172-31-42-235:~#`
- Line 27: `root@ip-172-31-42-235:~#`
- Line 28: `root@ip-172-31-42-235:~#`
- Line 29: `root@ip-172-31-42-235:~#`
- Line 30: `root@ip-172-31-42-235:~#`

Step 4: Use the username and password of the user to open MySQL shell.

Create database and create a table in that database.

```
29 ##### 1. Instance
28 Command to connect to your server, make sure you have key_pair.pem set to 400 permissions
27 ````bash
26 $ ssh -i "cc_aws_key.pem" [REDACTED]@amazonaws.com
25 ...
23 this takes you into aws server's shell and prompt
22
21
20 ##### 2. LAMP installation
19
18 LAMP stands for
17 1. L - Linux for Operating system
16 2. A - Apache HTTP server
15 3. M - mysql for relational database management system
14 4. P - php, perl or python as programming language
13
12 We installed are in our linux server, let's install other three on this server too
11 ````bash
10 $ sudo apt-get install apache2
9 $ sudo apt-get install mysql-server
8 $ sudo apt-get install php php-mysql
7 `````
6
5 Change permissions of /var/www/html/ folder
4 ````bash
3 $ sudo chmod 777 -R /var/www/html
2 `````
1
30 ##### 3. MySQL
1 Create user in MySQL
2
3 Use the username and password of the user to open MySQL shell.
4 Create database and create a table in that database.

ubuntu@ip-172-31-42-235:~$ mysql -u user2 -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE users_db;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near 'DATABSE users_db' at line 1
mysql> CREATE DATABASE users_db;
Query OK, 1 row affected (0.01 sec)

mysql> USE users_db;
Database changed
mysql> CREATE TABLE users(name VARCHAR(30));
Query OK, 0 rows affected (0.04 sec)

mysql> SELECT * FROM TABLE users
      -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near 'TABLE users' at line 1
mysql> SELECT * FROM users;
Empty set (0.01 sec)

mysql> ^D
ubuntu@ip-172-31-42-235:~$ 
```

4. PHP

Now we need to copy our files in server's /var/www/html directory from our local machine. For this we use `scp` command. It allows use to share files via ssh protocol

Step 5: Secure copy our files (index.html) in server's /var/www/html directory

```
$ scp -i key.pem local_file.php ubuntu@aws_server_dns.com:/var/www/html/
```

The screenshot shows a terminal window with two panes. The left pane displays a script for deploying a web application. The right pane shows the execution of the command and its output.

```
● ○○○
Konssole File Edit View Bookmarks Plugins Settings Help Search
~:bash - Konssole
08 Apr 2024 19:28
16
15 Change permissions of /var/www/html/ folder
14 ````bash
13 $ sudo chmod 777 -R /var/www/html
12 ``
11
10 ##### 3. MySQL
9 Create user in MySQL
8
7 Use the username and password of the use to open MySQL shell.
6 Create database and create a table in that database.
5
4 ##### 4. PHP
3 SCP(Secure Copy your PHP file(s) to the server).
2
1 ````bash
$ scp -i key.pem local_file.php ubuntu@aws_server_dns.com:/var/www/html

NORMAL note.md          utf-8 Y Δ Y markdown 97% 40:72
12 <!DOCTYPE html>
11 <html lang="en">
10   <head>
9     <title></title>
8     <meta charset="UTF-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <link href="css/style.css" rel="stylesheet">
5   </head>
4   <body>
3     <h1>THIS IS HOSTED ON AWS SERVER </h1>
2
1   </body>
13 </html>

index.html
"note.md" 41L, 1060B written
[server] 0:nvim*          13:7

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

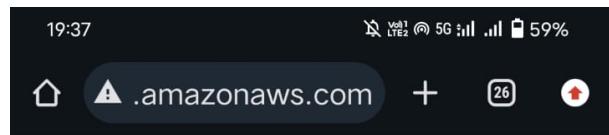
Last login: Mon Apr  8 11:45:21 2024 from 152.58.23.13
ubuntu@ip-172-31-42-235:~$ ls /var/www/html
index.html
ubuntu@ip-172-31-42-235:~$ ls
ubuntu@ip-172-31-42-235:~$ cd /var/www/html
ubuntu@ip-172-31-42-235:/var/www/html$ ls
index.html
ubuntu@ip-172-31-42-235:/var/www/html$ sudo mv index.html index.html.bak
ubuntu@ip-172-31-42-235:/var/www/html$ ls
index.html index.html.bak
ubuntu@ip-172-31-42-235:/var/www/html$ 

om_d@deblinu:~/College/CC/Assignments/03_assignment
> scp -i ~/cc_aws_key.pem index.html 'zonaws.com:/var/www/html/index.html'
index.html
100% 324      1.6KB/s  00:00
om_d@deblinu:~/College/CC/Assignments/03_assignment
>
```

And thus we have deployed our web-application to AWS cloud and we can access this by putting AWS instance's DNS link in browser



THIS IS HOSTED ON AWS SERVER



**THIS IS HOSTED ON
AWS SERVER**