

# ASSIGNMENT-1

## A. Create and Insert the following table

**Table Name- client \_master**  
**Description- Used to store client information**

Column No	Column Name	Data Type	Size	Attributes
1	Client_no	Varchar2	6	Primary key, first letter must start with 'C'
2	Name	Varchar2	30	Not NULL
3	Address1	Varchar2	30	
4	Address2	Varchar2	30	
5	City	Varchar2	15	
6	State	Varchar2	15	
7	Pincode	Number	6	
8	Balance_due	Number	10,2	

**Data of client\_master table**

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6	Col-7	Col-8
C001	Ivan Bayross	P-76	Worli	Bombay	Maharastra	400054	15000
C002	VandanaSatiwal	128	Adams Street	Madras	TamilNadu	780001	0
C003	PramadaJaguste	157	Gopalpur	Kolkata	West Bengal	700058	5000
C004	BasuNavindgi	A/12	Nariman	Bombay	Maharastra	400056	0
C005	Ravi Sreedharan	B/34	Rajnagar	Delhi	Delhi	100001	2000
C006	Rukmini	Q-12	Bandra	Bombay	Maharastra	400050	0

## SQL Commands:

```
create table client_master(
```

```
    Client_no varchar2(6) primary key check (Client_no like 'C%'),
```

```
    Name varchar2(30) NOT NULL,
```

```
    Address1 varchar2(30) ,
```

```
    Address2 varchar2(30) ,
```

```
    City varchar2(15) ,
```

```
    State varchar2(15) ,
```

```
    Pincode Number(6) ,
```

```
    Balance_due Number(10,2)
```

```
);
```

```
INSERT INTO client_master VALUES('C001','Ivan Bayross','P-76','Worli','Bombay','Maharastra',400054,15000);
```

```
INSERT INTO client_master VALUES('C002','Vandana Satiwal','128','Adams Street','Madras','Tamil Nadu',780001,0);
```

```
INSERT INTO client_master VALUES('C003','Pramada Jaguste','157','Gopalpur','Kolkata','West Bengal',700058,5000);
```

```
INSERT INTO client_master VALUES('C004','Basu Navindgi','A/12','Nariman','Bombay','Maharastra',400056,0);
```

```
INSERT INTO client_master VALUES('C005','Ravi Sreedharan','B/34','Rajnagar','Delhi','Delhi',100001,2000);
```

```
INSERT INTO client_master VALUES('C006','Rukmini','Q-12','Bandra','Bombay','Maharastra',400050,0);
```

## SQL Prompt Output:

```
SQL> @E:\CSE_30\Client
```

```
Table created.
```

```
1 row created.|
```

```
SQL> desc client_master
```

Name	Null?	Type
CLIENT_NO	NOT NULL	VARCHAR2(6)
NAME	NOT NULL	VARCHAR2(30)
ADDRESS1		VARCHAR2(30)
ADDRESS2		VARCHAR2(30)
CITY		VARCHAR2(15)
STATE		VARCHAR2(15)
PINCODE		NUMBER(6)
BALANCE_DUE		NUMBER(10,2)

```
SQL> set linesize 500
```

```
SQL> select * from client_master;
```

CLIENT_NAME	ADDRESS1	ADDRESS2	CITY	STATE	PINCODE	BALANCE_DUE
C001 Ivan Bayross	P-76	Worli	Bombay	Maharashtra	400054	15000
C002 Vandana Satiwal	128	Adams Street	Madras	Tamil Nadu	780001	0
C003 Pramada Jaguste	157	Gopalpur	Kolkata	West Bengal	70058	5000
C004 Basu Navindgi	A/12	Nariman	Bombay	Maharashtra	400056	0
C005 Ravi Sreedharan	B/34	Rajnagar	Delhi	Delhi	100001	2000
C006 Rukmini	Q-12	Bandra	Bombay	Maharashtra	400050	0

```
6 rows selected.
```

## **B. Create and Insert the following table**

**Table Name- product\_master:****Description- Used to store product information**

Column No	Column Name	Data Type	Size	Attributes
1	Product_no	Varchar2	6	Primary key, First letter must start with 'P'
2	Description	Varchar2	40	Not null
3	Profit_percent	Number	4,2	Not null
4	Unit_measure	Varchar2	10	Not null
5	Qty_on_hand	Number	8	Not null
6	Reorder_level	Number	8	Not null
7	Sell_price	Number	8,2	Not null, cannot be 0
8	Cost_price	Number	8,2	Not null, cannot be 0

**Data of product\_master table**

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6	Col-7	Col-8
P00001	1.44 Floppies	5	Piece	100	20	525	500
P03453	Monitors	6	Piece	10	3	12000	11280
P06734	Mouse	5	Piece	20	5	1050	1000
P07865	1.22 Floppies	5	Piece	100	20	525	500
P07868	Keyboard	2	Piece	10	3	3150	3050
P07885	CD Drive	2.5	Piece	10	3	5250	5100
P07965	540 HDD	4	Piece	10	3	8400	8000
P07975	1.44 Drive	5	Piece	10	3	1050	900
P08865	1.22 Drive	5	Piece	2	3	1025	850

## SQL Commands:

```
create table product_master(  
    Product_no varchar2(6) Primary key check (Product_no like 'P%'),  
    Description varchar2(40) NOT NULL,  
    Profit_percent number(4,2) not null ,  
    Unit_measure varchar2(10) not null ,  
    Qty_on_hand number(8) not null,  
    Reorder_level number(8) not null,  
    Sell_price number(8,2) not null check(Sell_price > 0),  
    Cost_price number(8,2) not null check(Cost_price > 0)  
);  
  
Insert into product_master values('P00001','1.44 Floppies',5,'Piece',100,20,525,500);  
Insert into product_master values('P03453','Monitors',6,'Piece',10,3,12000,11280);  
Insert into product_master values('P06734','Mouse',5,'Piece',20,5,1050,1000);  
Insert into product_master values('P07865','1.22 Floppies',5,'Piece',100,20,525,500);  
Insert into product_master values('P07868','Keyboard',2,'Piece',10,3,3150,3050);  
Insert into product_master values('P07885','CD Drive',2.5,'Piece',10,3,5250,5100);  
Insert into product_master values('P07965','540 HDD',4,'Piece',10,3,8400,8000);  
Insert into product_master values('P07975','1.44 Drive',5,'Piece',10,3,1050,900);  
Insert into product_master values('P08865','1.22 Drive',5,'Piece',2,3,1025,850);
```

## SQL Prompt Output:

## C. Create and Insert the following table

**Table Name- salesman\_master:**

**Description- Used to store salesman working for company**

Column No	Column Name	Data Type	Size	Attributes
1	Salesman_no	Varchar2	6	Primary key,first letter must start with 'S'
2	Salesman_name	Varchar2	30	Not null
3	Address1	Varchar2	30	Not null
4	Address2	Varchar2	30	
5	City	Varchar2	20	
6	Pincode	Number	8	
7	State	Varchar2	20	
8	Sal_amt	Number	8, 2	Not null, cannot be 0

**Data of salesman\_master table**

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6	Col-7	Col-8
S001	Kiran	A/14	Worli	Bombay	400002	Maharastra	3000
S002	Manish	65	Nariman	Bombay	400001	Maharastra	3000
S003	Ravi	P-7	Bandra	Bombay	400032	Maharastra	3000
S004	Asish	A/5	Juhu	Bombay	400044	Maharastra	3000

## SQL Commands:

```
CREATE table salesman_master(
```

```
    Salesman_No varchar2(6) PRIMARY key ,
```

```
    CHECK(Salesman_No like 'S%'),
```

```
    Salesman_name varchar2(30) not null,
```

```
    Address1 varchar2(30) not null,
```

```
    Address2 varchar2(30),
```

```
    City varchar2(20),
```

```
    Pincode NUMBER(8),
```

```
    State VARCHAR2(20),
```

```
    Sal_amt number(8,2) not null check(Sal_amt > 0)
```

```
);
```

```
insert into salesman_master values ('S001','Kiran','A/14','Worli','Bombay',400002,'Maharastra',3000);
```

```
insert into salesman_master values ('S002','Manish','65','Nariman','Bombay',400001,'Maharastra',3000);
```

```
insert into salesman_master values ('S003','Ravi','P-7','Bandra','Bombay',400032,'Maharastra',3000);
```

```
insert into salesman_master values ('S004','Asish','A/5','Juhu','Bombay',400044,'Maharastra',3000);
```

## SQL Prompt Output:

Name	Null?	Type
SALESMAN_NO	NOT NULL	VARCHAR2(6)
SALESMAN_NAME	NOT NULL	VARCHAR2(30)
ADDRESS1	NOT NULL	VARCHAR2(30)
ADDRESS2		VARCHAR2(30)
CITY		VARCHAR2(20)
PINCODE		NUMBER(8)
STATE		VARCHAR2(20)
SAL_AMT	NOT NULL	NUMBER(8,2)

SQL> set linesize 500;
SQL> select * from salesman_master;
SALESMALESM SALESMAN_NAME ADDRESS1 ADDRESS2 CITY PINCODE STATE SAL_AMT
S001 Kiran A/14 Worli Bombay 400002 Maharashtra 3000
S002 Manish 65 Nariman Bombay 400001 Maharashtra 3000
S003 Ravi P-7 Bandra Bombay 400032 Maharashtra 3000
S004 Asish A/5 Juhu Bombay 400044 Maharashtra 3000

## D. Create and Insert the following table

**Table Name- sales\_order:**  
**Description- Used to store client's orders**

Column No	Column Name	Data Type	Size	Attributes
1	Order_no	Varchar2	6	Primary key, first letter must start with 'O'
2	Order_date	Date		
3	Client_no	Varchar2	6	Foreign key references Client_master table
4	Salesman_no	Varchar2	6	Foreign key references salesman_master table
5	Delivery_type	Char	1	Delivery part(P),full(F) Default 'F'
6	Bill_y_n	Char	1	
7	Delivery_date	Date		Cannot be less than Order_date
8	Order_status	Varchar2	10	Values('InProcess', 'Fullfilled', 'BackOrder', 'Cancelled')

### Data of sales\_order table

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6	Col-7	Col-8
O19001	12-Jan-96	C001	S001	F	N	20-Jan-96	InProcess
O19002	25-Jan-96	C002	S002	P	N	27-Jan-96	BackOrder
O46865	18-Feb-96	C003	S003	F	Y	20-Feb-96	Fullfilled
O19003	03-Apr-96	C001	S001	F	Y	07-Apr-96	Fullfilled
O46866	20-May-96	C004	S002	P	N	22-May-96	Cancelled
O19008	24-May-96	C005	S004	F	N	26-May-96	InProcess

## SQL Commands:

```
create table sales_order(
    Order_No varchar2(6) PRIMARY key,
    check(Order_No like 'O%'),
    Order_date Date,
    Client_No varchar2(6) REFERENCES client_master(Client_No),
    Salesman_No varchar2(6) REFERENCES salesman_master(Salesman_No),
    Delivery_type Char(1) DEFAULT 'F' check(Delivery_type in ('P','F')),
    Bill_y_n char(1),
    Delivery_date date ,
    Order_status varchar2(10),
    constraint ck_order_status check(Order_status in ('InProcess','Fullfilled','BackOrder','Cancelled')),
    constraint ck_delivery_date check (Delivery_date >= Order_date)
);

insert into sales_order VALUES ('O19001','12-Jan-96','C001','S001','F','N','20-Jan-96','InProcess');
insert into sales_order VALUES ('O19002','25-Jan-96','C002','S002','P','N','27-Jan-96','BackOrder');
insert into sales_order VALUES ('O46865','18-Feb-96','C003','S003','F','Y','20-Feb-96','Fullfilled');
insert into sales_order VALUES ('O19003','03-Apr-96','C001','S001','F','Y','07-Apr-96','Fullfilled');
insert into sales_order VALUES ('O46866','20-May-96','C004','S002','P','N','22-May-96','Cancelled');
insert into sales_order VALUES ('O19008','24-May-96','C005','S004','F','N','26-May-96','InProcess');
```

## SQL Prompt Output:

```
SQL> @E:\CSE_30\Ass1\ass_1_4
Table created.

1 row created.
```

```
SQL> desc sales_order
Name          Null?    Type
-----        -----
ORDER_NO      NOT NULL VARCHAR2(6)
ORDER_DATE    DATE
CLIENT_NO     VARCHAR2(6)
SALESMAN_NO   VARCHAR2(6)
DELIVERY_TYPE CHAR(1)
BILL_Y_N      CHAR(1)
DELIVERY_DATE DATE
ORDER_STATUS  VARCHAR2(10)
```

```

SQL> set linesize 500;
SQL> select * from sales_order;

ORDER_ ORDER_DAT CLIENT SALESMD B DELIVERY_ ORDER_STAT
-----
019001 12-JAN-96 C001 S001 F N 20-JAN-96 InProcess
019002 25-JAN-96 C002 S002 P N 27-JAN-96 BackOrder
046865 18-FEB-96 C003 S003 F Y 20-FEB-96 Fullfilled
019003 03-APR-96 C001 S001 F Y 07-APR-96 Fullfilled
046866 20-MAY-96 C004 S002 P N 22-MAY-96 Cancelled
019008 24-MAY-96 C005 S004 F N 26-MAY-96 InProcess

6 rows selected.

```

## E. Create and Insert the following table

**Table Name- sales\_order\_details:**

**Description- Used to store client's orders with details of each product ordered**

Column No	Column Name	Data Type	Size	Attributes
1	Order_no	Varchar2	6	Foreign key references sales_order table
2	Product_no	Varchar2	6	Foreign key references product_master table
3	Qty_ordered	Number	8	
4	Qty_disp	Number	8	
5	Product_rate	Number	10, 2	

**Data of sales\_order\_details**

Col-1	Col-2	Col-3	Col-4	Col-5
O19001	P00001	4	4	525
O19001	P07965	2	1	8400
O19001	P07885	2	1	5250
O19002	P00001	10	0	525
O46865	P07868	3	3	3150
O46865	P07885	3	1	5250
O46865	P00001	10	10	525
O46865	P03453	4	4	1050
O19003	P03453	2	2	1050
O19003	P06734	1	1	12000
O46866	P07965	1	0	8400
O46866	P07975	1	0	1050
O19008	P00001	10	5	525
O19008	P07975	5	3	1050

## SQL Commands:

```
create table sales_order_details(
    Order_No varchar2(6) references sales_order(Order_No),
    Product_No varchar2(6) references product_master(Product_No),
    Qty_ordered NUMBER(8),
    Qty_disp number(8),
    Product_rate number(10,2)
);
desc sales_order_details;
insert into sales_order_details VALUES ('O19001','P00001',4,4,525);
insert into sales_order_details VALUES ('O19001','P07965',2,1,8400);
insert into sales_order_details VALUES ('O19001','P07885',2,1,5250);
insert into sales_order_details VALUES ('O19002','P00001',10,0,525);
insert into sales_order_details VALUES ('O46865','P07868',3,3,3150);
insert into sales_order_details VALUES ('O46865','P07885',3,1,5250);
insert into sales_order_details VALUES ('O46865','P00001',10,10,525);
insert into sales_order_details VALUES ('O46865','P03453',4,4,1050);
insert into sales_order_details VALUES ('O19003','P03453',2,2,1050);
insert into sales_order_details VALUES ('O19003','P06734',1,1,12000);
insert into sales_order_details VALUES ('O46866','P07965',1,0,8400);
insert into sales_order_details VALUES ('O46866','P07975',1,0,1050);
insert into sales_order_details VALUES ('O19008','P00001',10,5,525);
insert into sales_order_details VALUES ('O19008','P07975',5,3,1050);
```

## SQL Prompt Output:

```
SQL> desc sales_order_details
```

Name	Null?	Type
ORDER_NO		VARCHAR2(6)
PRODUCT_NO		VARCHAR2(6)
QTY_ORDERED		NUMBER(8)
QTY_DISP		NUMBER(8)
PRODUCT_RATE		NUMBER(10,2)

```
SQL> select * from sales_order_details;
```

```
ORDER_ PRODUC QTY_ORDERED QTY_DISP PRODUCT_RATE
```

019001	P00001	4	4	525
019001	P07965	2	1	8400
019001	P07885	2	1	5250
019002	P00001	10	0	525
046865	P07868	3	3	3150
046865	P07885	3	1	5250
046865	P00001	10	10	525
046865	P03453	4	4	1050
019003	P03453	2	2	1050
019003	P06734	1	1	12000
046866	P07965	1	0	8400

```
ORDER_ PRODUC QTY_ORDERED QTY_DISP PRODUCT_RATE
```

046866	P07975	1	0	1050
019008	P00001	10	5	525
019008	P07975	5	3	1050

```
14 rows selected.
```

```
SQL> spool off
```

# ASSIGNMENT-2

- 1. Find the names of all clients having ‘a’ as the second letter in their names.**

SQL Command:

```
select name from client_master where name like '_a%' ;
```

Output:

```
SQL> select name from client_master where name like '_a%' ;  
  
NAME  
-----  
Vandana Satiwal  
Basu Navindgi  
Ravi Sreedharan
```

- 2. Find out the clients who do not stay in a city whose first letter is ‘B’.**

SQL Command:

```
select name from client_master where city not like 'B%' ;
```

Output:

```
SQL> select name from client_master where city not like 'B%' ;  
  
NAME  
-----  
Vandana Satiwal  
Pramada Jaguste  
Ravi Sreedharan
```

- 3. List the names and city of all clients who have exactly 12 characters in length and starts with ‘I’.**

SQL Command:

```
select name, city from client_master where length(name)=12 and name like 'I%' ;
```

Output:

```
SQL> select name, city from client_master where length(name)=12 and name like 'I%' ;  
  
NAME          CITY  
-----  
Ivan Bayross      Bombay
```

- 4. Find the list of all clients who stay in ‘Bombay’ or ‘Delhi’.**

SQL Command:

```
select name from client_master where city='Bombay' or city='Delhi' ;
```

Output:

```
SQL> select name from client_master where city='Bombay' or city='Delhi';

NAME
-----
Ivan Bayross
Basu Navindgi
Ravi Sreedharan
Rukmini
```

**5. Print the list of all clients whose bal\_due is greater than value 10,000.**

SQL Command:

```
select name from client_master where balance_due>10000;
```

Output:

```
SQL> select name from client_master where balance_due>10000;

NAME
-----
Ivan Bayross
```

**6. Print the information from sales\_order table for orders places in the month of January.**

SQL Command:

```
select * from sales_order where to_char (order_date,'Mon')='Jan' ;
```

Output:

```
SQL> select * from sales_order where to_char (order_date,'Mon')='Jan' ;

ORDER_ ORDER_DAT CLIENT SALESM D B DELIVERY_ ORDER_STAT
----- -----
019001 12-JAN-96 C001    S001    F N 20-JAN-96 InProcess
019002 25-JAN-96 C002    S002    P N 27-JAN-96 BackOrder
```

**7. Display the order information for client\_no 'C001' and 'C002'.**

SQL Command:

```
select * from sales_order where client_no in ('C001', 'C002');
```

Output:

```
SQL> select * from sales_order where client_no in ('c001', 'c002');

ORDER_ ORDER_DAT CLIENT SALESM D B DELIVERY_ ORDER_STAT
----- -----
019001 12-JAN-96 C001    S001    F N 20-JAN-96 InProcess
019002 25-JAN-96 C002    S002    P N 27-JAN-96 BackOrder
019003 03-APR-96 C001    S001    F Y 07-APR-96 Fullfilled
```

## 8. Find products whose selling price greater than 2000 and less than 5000.

SQL Command:

```
select * from product_master where Sell_price>2000 and sell_price<5000;
```

Output:

```
SQL> select * from product_master where Sell_price>2000 and sell_price<5000;

PRODUC DESCRIPTION PROFIT_PERCENT UNIT_MEASU
----- QTY_ON_HAND REORDER_LEVEL SELL_PRICE COST_PRICE
----- P07868 Keyboard 2 Piece
| 10 | 3 | 3150 | 3050
```

## 9. Find products whose selling price is more than 1500.Calculate a new selling price as original selling price\*1.15. Rename the new column in the above query is New\_price.

SQL Command:

```
select Sell_price, Sell_price*1.15 New_Price from product_master;
```

Output:

```
SQL> select Sell_price, Sell_price*1.15 New_Price from product_master;

SELL_PRICE NEW_PRICE
----- -----
| 525 | 603.75
| 12000 | 13800
| 1050 | 1207.5
| 525 | 603.75
| 3150 | 3622.5
| 5250 | 6037.5
| 8400 | 9660
| 1050 | 1207.5
| 1025 | 1178.75

9 rows selected.
```

## 10. List the names, city and state of clients who are not in the state of 'Maharashtra'.

SQL Command:

```
select name, city, state from Client_master where State!='Maharashtra' ;
```

Output:

```
SQL> select name, city, state from Client_master where State!='Maharashtra' ;

NAME CITY STATE
----- -----
Ivan Bayross Bombay Maharashtra
Vandana Satiwal Madras Tamil Nadu
Pramada Jaguste Kolkata West Bengal
Basu Navindgi Bombay Maharashtra
Ravi Sreedharan Delhi Delhi
Rukmini Bombay Maharashtra

6 rows selected.
```

## **11. Display the month (in alphabets) and date when the order must be delivered.**

SQL Command:

```
select to_char(delivery_date, 'Month-dd') from Sales_order;
```

Output:

```
SQL> select to_char(delivery_date, 'Month-dd') from Sales_order;
TO_CHAR(DELIVERY_DATE, 'MONTH-DD')
-----
January -20
January -27
February -20
April -07
May -22
May -26

6 rows selected.
```

## **12. Display the Order\_date in the format 'DD-Month-YY' e.g., 12-February-13.**

SQL Command:

```
select to_char(order_date, 'DD-Month-YY') O_date from Sales_order;
```

Output:

```
153
154 SQL> select to_char(order_date, 'DD-Month-YY') O_date from Sales_order;
155
156 O_DATE
157 -----
158 12-January -96
159 25-January -96
160 18-February -96
161 03-April -96
162 20-May -96
163 24-May -96
164
165 6 rows selected.
166
```

## **13. Find the date, 15 days after today's date**

SQL Command:

```
select sysdate+15 New_date from dual;
```

Output:

```
SQL> select sysdate+15 New_date from dual;
NEW_DATE
-----
15-MAR-23
```

# **ASSIGNMENT-3**

## **1. Count the total number of orders.**

SQL Command:

```
select count(*) from sales_order;
```

Output:

```
SQL> select count(*) from sales_order;
      COUNT(*)
-----
      6
```

## **2. Calculate the average price of all the products.**

SQL Command:

```
select avg(sell_price) from product_master;
select avg(cost_price) from product_master;
```

Output:

```
SQL> select avg(sell_price) from product_master
  2  ;
      AVG(SELL_PRICE)
-----
      3663.88889

SQL> select avg(cost_price) from product_master;

      AVG(COST_PRICE)
-----
      3464.44444
```

## **3. Count the number of products having price greater than or equal to 1500.**

SQL Command:

```
select count(*) from product_master where (cost_price>1500);
```

Output:

```
SQL> select count(*) from product_master where (cost_price>1500);
      COUNT(*)
-----
      4
```

#### **4. Determine the maximum and minimum product prices. Rename the output as max\_price and min\_price respectively.**

SQL Command:

```
select min(cost_price) MIN_PRICE,max(cost_price) MAX_PRICE from product_master;
```

Output:

```
SQL> select min(cost_price) MIN_PRICE,max(cost_price) MAX_PRICE from product_master;

MIN_PRICE    MAX_PRICE
-----      -----
      500        11280
```

#### **5. Change the City of the Client\_no 'C005' to 'Madras'.**

SQL Command:

```
update client_master set city='MADRAS' where client_no='C005';
```

Output:

```
SQL> update client_master set city='MADRAS' where client_no='C005'
  2 ;
1 row updated.

SQL> select * from client_master
  2 ;

CLIENT NAME          ADDRESS1          ADDRESS2          CITY          STATE          PINCODE      BALANCE_DUE
-----      -----
C001  Ivan Bayross      P-76            Worli          Bombay        Maharastra   400054      15000
C002  Vandana Satiwal    128            Adams Street    Madras        Tamil Nadu   780001      0
C003  Pramada Jaguste     157            Gopalpur       Kolkata       Weste Bengal 70058       5000
C004  Basu Navindgi      A/12           Nariman         Bombay        Maharastra   400056      0
C005  Ravi Sreedharan     B/34           Rajnagar       MADRAS        Delhi        100001      2000
C006  Rukmini              Q-12           Bandra          Bombay        Maharastra   400050      0
6 rows selected.
```

#### **6. Change the Bal\_due of Client\_no 'C005' to Rs.3000/-.**

SQL Command:

```
update client_master set balance_due='3000' where client_no='C005';
```

Output:

```
SQL> update client_master set balance_due='3000' where client_no='C005';
1 row updated.

SQL> select * from client_master
  2 ;

CLIENT NAME          ADDRESS1          ADDRESS2          CITY          STATE          PINCODE      BALANCE_DUE
-----      -----
C001  Ivan Bayross      P-76            Worli          Bombay        Maharastra   400054      15000
C002  Vandana Satiwal    128            Adams Street    Madras        Tamil Nadu   780001      0
C003  Pramada Jaguste     157            Gopalpur       Kolkata       Weste Bengal 70058       5000
C004  Basu Navindgi      A/12           Nariman         Bombay        Maharastra   400056      0
C005  Ravi Sreedharan     B/34           Rajnagar       MADRAS        Delhi        100001      3000
C006  Rukmini              Q-12           Bandra          Bombay        Maharastra   400050      0
6 rows selected.
```

## **7. Delete from client\_master where the column state holds the value ‘Tamil Nadu’.**

SQL Command:

```
delete from client_master where state='Tamil Nadu';
```

Output:

```
SQL> delete from client_master where state='Tamil Nadu';
delete from client_master where state='Tamil Nadu'
*
ERROR at line 1:
ORA-02292: integrity constraint (CSE30.SYS_C00102035) violated - child record found
```

## **8. Add a column called ‘Telephone’ of data type ‘number’ and size 10 in the table client\_master.**

SQL Command:

```
Alter table client_master add(Telephone number(10));
```

Output:

```
SQL> Alter table client_master add(Telephone number(10));
Table altered.

SQL> select * from client_master
  2  ;

CLIENT NAME          ADDRESS1           ADDRESS2           CITY      STATE        PINCODE  BALANCE_DUE  TELEPHONE
-----  -----
C001  Ivan Bayross    P-76              Worli            Bombay   Maharashtra  400054    15000
C002  Vandana Satiwal 128              Adams Street     Madras   Tamil Nadu    780001    0
C003  Pramada Jaguste 157              Gopalpur        Kolkata  Weste Bengal  70058     5000
C004  Basu Navindgi   A/12             Nariman         Bombay   Maharashtra  400056    0
C005  Ravi Sreedharan B/34             Rajnagar        MADRAS   Delhi        100001    3000
C006  Rukmini          Q-12             Bandra          Bombay   Maharashtra  400050    0

6 rows selected.
```

## **9. Change the size of data type Pin\_code to 10 in the table client\_master**

SQL Command:

```
Alter table client_master modify(pincode number(10));
```

Output:

```
SQL> Alter table client_master modify(pincode number(10));
Table altered.

SQL> select * from client_master
  2  ;

CLIENT NAME          ADDRESS1           ADDRESS2           CITY      STATE        PINCODE  BALANCE_DUE  TELEPHONE
-----  -----
C001  Ivan Bayross    P-76              Worli            Bombay   Maharashtra  400054    15000
C002  Vandana Satiwal 128              Adams street     Madras   Tamil Nadu    780001    0
C003  Pramada Jaguste 157              Gopalpur        Kolkata  Weste Bengal  70058     5000
C004  Basu Navindgi   A/12             Nariman         Bombay   Maharashtra  400056    0
C005  Ravi Sreedharan B/34             Rajnagar        MADRAS   Delhi        100001    3000
C006  Rukmini          Q-12             Bandra          Bombay   Maharashtra  400050    0

6 rows selected.

SQL> desc client_master;
 Name          Type
 Null?         Type
-----  -----
CLIENT NO      NOT NULL, VARCHAR2(6)
NAME           NOT NULL, VARCHAR2(30)
ADDRESS1        VARCHAR2(30)
ADDRESS2        VARCHAR2(30)
CITY           VARCHAR2(15)
STATE          VARCHAR2(15)
PINCODE        NUMBER(10)
BALANCE_DUE    NUMBER(10,2)
TELEPHONE      
```

## 10. Drop the column Address2 from the table client\_master

SQL Command:

```
Alter table client_master drop(address2);
```

Output:

```
SQL> Alter table client_master drop(address2);
Table altered.

SQL> desc client_master;
Name          Null?    Type
-----        -----
CLIENT_NO      NOT NULL VARCHAR2(6)
NAME           NOT NULL VARCHAR2(30)
ADDRESS1        VARCHAR2(30)
CITY            VARCHAR2(15)
STATE           VARCHAR2(15)
PINCODE         NUMBER(10)
BALANCE_DUE    NUMBER(10,2)
TELEPHONE      NUMBER(10)

SQL> select * from client_master
2 ;
-----+-----+-----+-----+-----+-----+-----+-----+
CLIENT_NAME    ADDRESS1      CITY       STATE      PINCODE  BALANCE_DUE  TELEPHONE
-----+-----+-----+-----+-----+-----+-----+-----+
C001   Ivan Bayross     P-76        Bombay    Maharastra 400054      15000
C002   Vandana Satiwal  128        Madras    Tamil Nadu 780001          0
C003   Pramada Jaguste 157        Kolkata  Weste Bengal 70058       5000
C004   Basu Navindgi   A/12       Bombay    Maharastra 400056          0
C005   Ravi Sreedharan B/34       MADRAS    Delhi      100001      3000
C006   Rukmini          Q-12       Bombay    Maharastra 400050          0
-----+-----+-----+-----+-----+-----+-----+-----+
6 rows selected.
```

## 11. Create another table client\_master\_duplicate with the same structure of client\_master (without copying the data of the table client\_master).

SQL Command:

```
create table client_master_duplicate as select * from client_master where 1=2;
```

Output:

```
SQL> create table client_master_duplicate as select * from client_master where 1=2;
Table created.

SQL> select * from client_master_duplicate;
no rows selected

SQL> desc client_master_duplicate;
Name          Null?    Type
-----        -----
CLIENT_NO      NOT NULL VARCHAR2(6)
NAME           NOT NULL VARCHAR2(30)
ADDRESS1        VARCHAR2(30)
CITY            VARCHAR2(15)
STATE           VARCHAR2(15)
PINCODE         NUMBER(10)
BALANCE_DUE    NUMBER(10,2)
TELEPHONE      NUMBER(10)

SQL> insert into client_master_duplicate select * from client_master;
6 rows created.
```

## **12. Insert the data into client\_master\_duplicate table from client\_master table.**

SQL Command:

```
insert into client_master_duplicate select * from client_master;
```

Output:

```
SQL> insert into client_master_duplicate select * from client_master;
6 rows created.
```

CLIENT NAME	ADDRESS1	CITY	STATE	PINCODE	BALANCE_DUE	TELEPHONE
C001 Ivan Bayross	P-76	Bombay	Maharashtra	400054	15000	
C002 Vandana Satiwal	128	Madras	TamilNadu	780001	0	
C003 Pramada Jaguste	157	Kolkata	West Bengal	700058	5000	
C004 Basu Navindgi	A/12	Bombay	Maharashtra	400056	0	
C005 Ravi Sreedharan	B/34	Madras	Delhi	100001	3000	
C006 Rukmini	Q-12	Bombay	Maharashtra	400050	0	

6 rows selected.

## **13. Rename the table client\_master\_duplicate to c\_master.**

SQL Command:

```
rename client_master_duplicate to c_master;
```

Output:

```
SQL> rename client_master_duplicate to c_master;
Table renamed.

SQL> select * from tab;

TNAME          TABTYPE  CLUSTERID
-----
STUDENTTABLE    TABLE
STUDENT         TABLE
STD             TABLE
SALES_ORDER_DETAILS TABLE
SALES_ORDER     TABLE
SALESMAN_MASTER TABLE
PRODUCT_MASTER  TABLE
DEPT            TABLE
DATAFLAIR       TABLE
C_MASTER        TABLE
CLIENT_MASTER   TABLE

11 rows selected.
```

## **14. Destroy the table c\_master with its data.**

SQL Command:

```
drop table c_master;
```

Output:

```
SQL> drop table c_master;
Table dropped.

SQL> select * from tab;

TNAME          TABTYPE  CLUSTERID
-----
STUDENTTABLE    TABLE
STUDENT         TABLE
STD             TABLE
SALES_ORDER_DETAILS TABLE
SALES_ORDER     TABLE
SALESMAN_MASTER TABLE
PRODUCT_MASTER  TABLE
DEPT            TABLE
DATAFLAIR       TABLE
CLIENT_MASTER   TABLE
BIN$EOi+UX8XQo6Qa5U/AetZnA==$0 TABLE

11 rows selected.
```

# **ASSIGNMENT-4**

## **Dept Table Creation and Insertion:**

```
create table dept(  
    deptno number(2,0),  
    dname varchar2(14),  
    loc  varchar2(13),  
    constraint pk_dept primary key (deptno)  
);
```

```
insert into dept  
values(10, 'ACCOUNTING', 'NEW YORK');  
insert into dept  
values(20, 'RESEARCH', 'DALLAS');  
insert into dept  
values(30, 'SALES', 'CHICAGO');  
insert into dept  
values(40, 'OPERATIONS', 'BOSTON');
```

## **Output:**

DEPTNO DNAME LOC		
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

## **Emp Table Creation and Insertion:**

```
create table emp(  
    empno number(4,0),  
    ename varchar2(10),  
    job varchar2(9),  
    mgr number(4,0),  
    hiredate date,  
    sal number(7,2),  
    comm number(7,2),  
    deptno number(2,0),
```

```
constraint pk_emp primary key (empno),
constraint fk_deptno foreign key (deptno) references dept (deptno)

);

insert into emp
values(7839, 'KING', 'PRESIDENT', null, to_date('17-11-1981','dd-mm-yyyy'), 5000, null, 10);

insert into emp
values(7698, 'BLAKE', 'MANAGER', 7839, to_date('1-5-1981','dd-mm-yyyy'), 2850, null, 30);

insert into emp
values(7782, 'CLARK', 'MANAGER', 7839, to_date('9-6-1981','dd-mm-yyyy'), 2450, null, 10);

insert into emp
values(7566, 'JONES', 'MANAGER', 7839, to_date('2-4-1981','dd-mm-yyyy'), 2975, null, 20);

insert into emp
values(7788, 'SCOTT', 'ANALYST', 7566, to_date('13-JUL-87','dd-mm-rr') - 85, 3000, null, 20);

insert into emp
values(7902, 'FORD', 'ANALYST', 7566, to_date('3-12-1981','dd-mm-yyyy'), 3000, null, 20);

insert into emp
values(7369, 'SMITH', 'CLERK', 7902, to_date('17-12-1980','dd-mm-yyyy'), 800, null, 20);

insert into emp
values(7499, 'ALLEN', 'SALESMAN', 7698, to_date('20-2-1981','dd-mm-yyyy'), 1600, 300, 30);

insert into emp
values(7521, 'WARD', 'SALESMAN', 7698, to_date('22-2-1981','dd-mm-yyyy'), 1250, 500, 30);

insert into emp
values(7654, 'MARTIN', 'SALESMAN', 7698, to_date('28-9-1981','dd-mm-yyyy'), 1250, 1400, 30);

insert into emp
values(7844, 'TURNER', 'SALESMAN', 7698, to_date('8-9-1981','dd-mm-yyyy'), 1500, 0, 30);

insert into emp
values(7876, 'ADAMS', 'CLERK', 7788, to_date('13-JUL-87','dd-mm-yyyy'), 1100, null, 20);

insert into emp
values(7900, 'JAMES', 'CLERK', 7698, to_date('3-12-1981','dd-mm-yyyy'), 950, null, 30);

insert into emp
values(7934, 'MILLER', 'CLERK', 7782, to_date('23-1-1982','dd-mm-yyyy'), 1300, null, 10);
```

### Output:

```
SQL> select * from emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	13-JUL-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

14 rows selected.

### **1. Display the names of all employees' right aligning them to 15 characters.**

#### SQL Command:

```
select lpad(empno,15) from emp;
```

### Output:

```
LPAD(EMPNO,15)
```

	7369
	7499
	7521
	7566
	7654
	7698
	7782
	7788
	7839
	7844
	7876

### **2. Display the names of all employees' padding them to the right up to 15 characters with '\*'.**

#### SQL Command:

```
select lpad(empno,15,'*') from emp;
```

### Output:

```
LPAD(EMPNO,15,'*')
```

*****7369
*****7499
*****7521
*****7566
*****7654
*****7698
*****7782
*****7788
*****7839
*****7844
*****7876

- 3. Find the details of all the managers in department 10 and all clerks in department 20 and all employees who are neither managers nor clerks but whose salary is more than or equal to 2000/-.**

SQL Command:

```
select * from emp where (job='MANAGER' and deptno=10) or (job='CLERK' and deptno=20) or (job!='MANAGER' and job != 'CLERK' and sal>=2000);
```

Output:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7876	ADAMS	CLERK	7788	13-JUL-87	1100		20

6 rows selected.

- 4. List all the employees who have joined between 01/02/81 and 31/08/81.**

SQL Command:

```
select ename,hiredate from emp where hiredate between '1-FEB-81' and '31-AUG-81';
```

Output:

ENAME	HIREDATE
BLAKE	01-MAY-81
CLARK	09-JUN-81
JONES	02-APR-81
ALLEN	20-FEB-81
WARD	22-FEB-81

- 5. List all the employees who were joined as manager during 1981.**

SQL Command:

```
select ename,hiredate,job from emp where job='MANAGER' and to_char(hiredate,'YY')=81;
```

Output:

ENAME	HIREDATE	JOB
BLAKE	01-MAY-81	MANAGER
CLARK	09-JUN-81	MANAGER
JONES	02-APR-81	MANAGER

**6. List the employees whose salaries are 800, 1600 or 2450.**

SQL Command:

```
select ename, sal from emp where sal in (800,1600,2450);
```

Output:

ENAME	SAL
CLARK	2450
SMITH	800
ALLEN	1600

**7. List the names of all employees who are either ‘clerks’ or ‘salesman’ or ‘analyst’.**

SQL Command:

```
select ename, job from emp where job in ('MANAGER','CLERK','ANALYST');
```

Output:

ENAME	JOB
BLAKE	MANAGER
CLARK	MANAGER
JONES	MANAGER
SCOTT	ANALYST
FORD	ANALYST
SMITH	CLERK
ADAMS	CLERK
JAMES	CLERK
MILLER	CLERK

**8. List the total number of employees and the average salaries of the different departments.**

SQL Command:

```
select count(ename), avg(sal) from emp group by deptno;
```

Output:

COUNT(ENAME)	AVG(SAL)
6	1566.66667
5	2175
3	2916.66667

**9. Calculate the average salary of all employees whose department is 30.**

SQL Command:

```
select avg(sal) from emp where deptno=30;
```

Output:

AVG(SAL)
1566.66667

**10. Calculate the minimum salary earn by clerks.**

SQL Command:

```
select min(sal) from emp where job='CLERK';
```

Output:

MIN(SAL)
800

**11. Calculate the maximum salary earn by salesmen.**

SQL Command:

```
select max(sal) from emp where job='SALESMAN';
```

Output:

MAX(SAL)
1600

**12. -----**

**13. Calculate the no. of employees who are not getting any commission.**

SQL Command:

```
select count(ename) from emp where comm is NULL;
```

Output:

COUNT(ENAME)
10

#### **14. Find the department is not having any employee.**

SQL Command:

```
select deptno from emp;  
select dname from dept where deptno not in (10,20,30);
```

Output:

DEPTNO
10
30
10
20
20
20
30
30
30
30
DEPTNO
20
30
10

SQL> select dname from dept where deptno not in (10,20,30);
DNAME
OPERATIONS

# ASSIGNMENT-5

- 1. List all the employee names, dept name and the city, in department name order.**

SQL command:

```
select ename,dname,loc from emp,dept where emp.deptno=dept.deptno order by dname;
```

Output:

```
SQL> select ename,dname,loc from emp,dept where emp.deptno=dept.deptno order by dname;
-----+
ENAME    DNAME      LOC
-----+
CLARK    ACCOUNTING NEW YORK
MILLER   ACCOUNTING NEW YORK
KING     ACCOUNTING NEW YORK
FORD     RESEARCH   DALLAS
SCOTT    RESEARCH   DALLAS
JONES    RESEARCH   DALLAS
SMITH    RESEARCH   DALLAS
ADAMS   RESEARCH   DALLAS
WARD    SALES     CHICAGO
MARTIN  SALES     CHICAGO
TURNER  SALES     CHICAGO
-----+
ENAME    DNAME      LOC
-----+
JAMES   SALES     CHICAGO
ALLEN   SALES     CHICAGO
BLAKE   SALES     CHICAGO
-----+
14 rows selected.
```

- 2. List all employees working in Dallas in descending order of salary.**

SQL command:

```
select * from emp where deptno in (select deptno from dept where loc='DALLAS') order by sal desc;
```

Output:

```
SQL> select * from emp where deptno in (select deptno from dept where loc='DALLAS') order by sal desc;
-----+
EMPNO  ENAME      JOB          MGR HIREDATE    SAL     COMM    DEPTNO
-----+
7902    FORD       ANALYST     7566 03-DEC-81  3000    20
7788    SCOTT      ANALYST     7566 19-APR-87  3000    20
7566    JONES      MANAGER    7839 02-APR-81  2975    20
7876    ADAMS      CLERK      7788 13-JUL-87  1100    20
7369    SMITH      CLERK      7902 17-DEC-80   800     20
-----+
```

- 3. List employee name, department name, job and location of all employees who work in DALLAS.**

SQL command:

```
select ename,dname,job,loc from emp,dept where emp.deptno=dept.deptno and loc='DALLAS';
```

Output:

```
SQL> select ename,dname,job,loc from emp,dept where emp.deptno=dept.deptno and loc='DALLAS';
-----+
ENAME    DNAME      JOB          LOC
-----+
JONES   RESEARCH   MANAGER     DALLAS
SCOTT   RESEARCH   ANALYST    DALLAS
FORD    RESEARCH   ANALYST    DALLAS
SMITH   RESEARCH   CLERK      DALLAS
ADAMS   RESEARCH   CLERK      DALLAS
-----+
```

- 4. List the employee name, salary, PF, HRA, DA and gross salary; order the result in ascending order of gross. PF is 10% of salary, HRA is 60% of salary and DA is 40% of salary.**

SQL command:

```
select ename,sal,sal*0.1 PF,sal*0.6 HRA, sal*0.4 DA,(sal*0.1+sal*0.6+sal*0.4+sal) Gross_sal from emp order by Gross_sal;
```

Output:

```
SQL> select ename,sal,sal*0.1 PF,sal*0.6 HRA, sal*0.4 DA,(sal*0.1+sal*0.6+sal*0.4+sal) Gross_sal from emp order by Gross_sal;


| ENAME  | SAL  | PF    | HRA  | DA   | GROSS_SAL |
|--------|------|-------|------|------|-----------|
| SMITH  | 800  | 80    | 480  | 320  | 1680      |
| JAMES  | 950  | 95    | 570  | 380  | 1995      |
| ADAMS  | 1100 | 110   | 660  | 440  | 2310      |
| MARTIN | 1250 | 125   | 750  | 500  | 2625      |
| WARD   | 1250 | 125   | 750  | 500  | 2625      |
| MILLER | 1300 | 130   | 780  | 520  | 2730      |
| TURNER | 1500 | 150   | 900  | 600  | 3150      |
| ALLEN  | 1600 | 160   | 960  | 640  | 3360      |
| CLARK  | 2450 | 245   | 1470 | 980  | 5145      |
| BLAKE  | 2850 | 285   | 1710 | 1140 | 5985      |
| JONES  | 2975 | 297.5 | 1785 | 1190 | 6247.5    |
| ENAME  | SAL  | PF    | HRA  | DA   | GROSS_SAL |
| FORD   | 3000 | 300   | 1800 | 1200 | 6300      |
| SCOTT  | 3000 | 300   | 1800 | 1200 | 6300      |
| KING   | 5000 | 500   | 3000 | 2000 | 10500     |



14 rows selected.


```

## 5. Display names and salary of all the employees who report to KING.

### SQL command:

```
select ename,sal from emp where mgr=(select empno from emp where ename='KING');
```

### Output:

```
SQL> select ename,sal from emp where mgr=(select empno from emp where ename='KING');



| ENAME | SAL  |
|-------|------|
| BLAKE | 2850 |
| CLARK | 2450 |
| JONES | 2975 |


```

## 6. List all employees who work in DALLAS and earn more than any employee working in Chicago.

### SQL command:

```
select ename, loc, sal from emp, dept where emp.deptno=dept.deptno and loc = 'DALLAS' and sal >(select max(sal) from emp, dept where loc= 'CHICAGO' and emp.deptno=dept.deptno);
```

### Output:

```
SQL> select ename, loc, sal from emp, dept where emp.deptno=dept.deptno and loc = 'DALLAS' and sal >(select max(sal) from emp, dept where loc= 'CHICAGO' and emp.deptno=dept.deptno);



| ENAME | LOC    | SAL  |
|-------|--------|------|
| JONES | DALLAS | 2975 |
| SCOTT | DALLAS | 3000 |
| FORD  | DALLAS | 3000 |


```

## 7. List all employees who work in the same post as Smith.

### SQL command:

```
Select ename, job from emp where job=(select job from emp where ename= 'SMITH');
```

### Output:

```
SQL> Select ename, job from emp where job=(select job from emp where ename= 'SMITH');



| ENAME  | JOB   |
|--------|-------|
| SMITH  | CLERK |
| ADAMS  | CLERK |
| JAMES  | CLERK |
| MILLER | CLERK |


```

## 8. Find the job with the highest average salary.

SQL command:

```
Select job from emp where sal = (select max(avg(sal)) from emp group by job);
```

Output:

```
SQL> Select job from emp where sal = (select max(avg(sal)) from emp group by job);
JOB
-----
PRESIDENT
```

**9. List the top 10 earners in the company.**

SQL command:

```
Select ename, sal from (select ename, sal from emp order by sal desc) where rownum <=10;
```

Output:

```
SQL> Select ename, sal from (select ename, sal from emp order by sal desc) where rownum <=10;
ENAME          SAL
-----
KING           5000
SCOTT          3000
FORD            3000
JONES           2975
BLAKE           2850
CLARK           2450
ALLEN           1600
TURNER          1500
MILLER          1300
WARD             1250

10 rows selected.
```

**10. Display the names of all employees' replacing 'A' with 'a'.**

SQL command:

```
Select replace (ename, 'A', 'a') from emp;
```

Output:

```
SQL> Select replace (ename, 'A', 'a') from emp;
REPLACE(EN
-----
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
aLLEN
WaRD
MaRTIN
TURNER

REPLACE(EN
-----
aDaMS
JaMES
MILLER

14 rows selected.
```

**11. Show the salary of all the employees rounding it to the nearest Rs.1000/-.**

SQL command:

Select ename, sal, round (sal, -3) from emp;

Output:

```
SQL> Select ename, sal, round (sal, -3) from emp;

ENAME          SAL ROUND(SAL,-3)
-----  -----
KING           5000      5000
BLAKE          2850      3000
CLARK          2450      2000
JONES          2975      3000
SCOTT          3000      3000
FORD            3000      3000
SMITH           800       1000
ALLEN          1600      2000
WARD            1250      1000
MARTIN          1250      1000
TURNER          1500      2000

ENAME          SAL ROUND(SAL,-3)
-----  -----
ADAMS          1100      1000
JAMES           950       1000
MILLER          1300      1000

14 rows selected.
```

**12. Show the first three and last three characters of the names of all the employees.**

SQL command:

Select substr(ename,1,3), substr(ename, -3) from emp;

Output:

```
SQL> Select substr(ename,1,3), substr(ename, -3) from emp;

SUBSTR(ENAME SUBSTR(ENAME
-----  -----
KIN           ING
BLA           AKE
CLA           ARK
JON           NES
SCO           OTT
FOR           ORD
SMI           ITH
ALL           LEN
WAR           ARD
MAR           TIN
TUR           NER

SUBSTR(ENAME SUBSTR(ENAME
-----  -----
ADA           AMS
JAM           MES
MIL           LER

14 rows selected.
```

# **ASSIGNMENT – 6**

**Table: Client\_master**

Column_Name	Data type	Size	Attributes
Client_no	Varchar2	8	Primary Key
Name	Varchar2	20	Not Null
Address1	Varchar2	20	Not Null
Address2	Varchar2	20	
City	Varchar2	15	
State	Varchar2	15	
Pincode	Varchar2	8	
Bal_due	Number	8,3	

**1. Create a view vw\_client\_master using Client\_no, Name, Address1 and Bal\_due**

- a. Insert at least 3 records to vw\_client\_master.**
- b. Update a record to vw\_client\_master.**
- c. Delete a record from vw\_client\_master.**

**And check that the due to the above operation if the base table is affected or not.**

**SQL COMMANDS:**

```
create view vw_client_master as select client_no, Name, Address1, Balance_due from client_master;

insert into vw_client_master values ('C007', 'ABHISEK SINGH', 'BALLY', 15000);

insert into vw_client_master values ('C008', 'DIPAK DAS', 'SALTLAKE', 10000);

insert into vw_client_master values ('C009', 'RAMAN GUPTA', 'HOWRAH', 20000);

Update vw_client_master set Balance_due=8000 where client_no = 'C008';

Delete from vw_client_master where client_no = 'C007';
```

```
select * from vw_client_master;
```

```
select * from client_master;
```

**OUTPUT:**

```
SQL> create view vw_client_master as select client_no, Name, Address1, Balance_due from client_master;
View created.

SQL> insert into vw_client_master values ('C007', 'ABHISEK SINGH', 'BALLY', 15000);
1 row created.

SQL> insert into vw_client_master values ('C008', 'DIPAK DAS', 'SALTLAKE', 10000);
1 row created.

SQL> insert into vw_client_master values ('C009', 'RAMAN GUPTA', 'HOWRAH', 20000);
1 row created.

SQL> Update vw_client_master set Balance_due=8000 where client_no = 'C008';
1 row updated.

SQL> Delete from vw_client_master where client_no = 'C007';
1 row deleted.
```

```
SQL> select * from vw_client_master;
```

CLIENT NAME	ADDRESS1	BALANCE_DUE
C008 DIPAK DAS	SALT LAKE	8000
C009 RAMAN GUPTA	HOWRAH	20000
C001 Ivan Bayross	P-76	15000
C002 Vandana Satiwal	128	0
C003 Pramada Jaguste	157	5000
C004 Basu Navindgi	A/12	0
C005 Ravi Sreedharan	B/34	3000
C006 Rukmini	Q-12	0

```
8 rows selected.
```

```
SQL> set linesize 500;
```

```
SQL> select * from client_master;
```

CLIENT NAME	ADDRESS1	CITY	STATE	PINCODE	BALANCE_DUE	TELEPHONE
C008 DIPAK DAS	SALT LAKE				8000	
C009 RAMAN GUPTA	HOWRAH				20000	
C001 Ivan Bayross	P-76	Bombay	Maharashtra	400054	15000	
C002 Vandana Satiwal	128	Madras	Tamil Nadu	780001	0	
C003 Pramada Jaguste	157	Kolkata	West Bengal	700058	5000	
C004 Basu Navindgi	A/12	Bombay	Maharashtra	400056	0	
C005 Ravi Sreedharan	B/34	MADRAS	Delhi	100001	3000	
C006 Rukmini	Q-12	Bombay	Maharashtra	400050	0	

```
8 rows selected.
```

2. Create a view Vw\_sales\_det using Client\_no, Order\_no, Order\_date, Product\_no, Qty\_ordered, and order\_status for all order which have already marked as 'Backorder'. (Using the tables sales\_order, sales\_order\_details).

- Insert a record to vw\_sales\_det.
- Update the client\_no for a particular order\_no.
- Delete a record.
- Remove the views from database.

#### SQL COMMANDS:

```
create view vw_sales_det as select s1.client_no, s1.order_no, s1.order_date, s2.product_no, s2.qty_ordered, s1.order_status from sales_order s1, sales_order_details s2 where s1.order_no=s2.order_no and s1.order_status = 'BackOrder';
```

```
insert into vw_sales_det values ('C007', 'O19251', '12-Jan-91', 'P00091', 100, 'InProcess');
```

```
update vw_sales_det set client_no = 'C006' where order_no = 'O19002';
```

```
select * from vw_sales_det;
```

```
delete from vw_sales_det where client_no= 'C005';
```

```
select * from vw_sales_det;
```

```
drop view vw_sales_det;
```

## OUTPUT:

```
SQL> create view vw_sales_det as select s1.client_no, s1.order_no, s1.order_date, s2.product_no, s2.qty_ordered, s1.order_status from sales_order s1, sales_order_details s2 where s1.order_no=s2.order_no and s1.order_status = 'BackOrder';

View created.

SQL> insert into vw_sales_det values ('C007', '019251', '12-Jan-91', 'P00091', 100, 'InProcess');
insert into vw_sales_det values ('C007', '019251', '12-Jan-91', 'P00091', 100, 'InProcess')
*
ERROR at line 1:
ORA-01779: cannot modify a column which maps to a non key-preserved table

SQL> update vw_sales_det set client_no = 'C006' where order_no = '019002';
update vw_sales_det set client_no = 'C006' where order_no = '019002'
*
ERROR at line 1:
ORA-01779: cannot modify a column which maps to a non key-preserved table
```

```
SQL> select * from vw_sales_det;

CLIENT ORDER_ ORDER_DAT PRODUC QTY_ORDERED ORDER_STAT
----- ----- -----
C002    019002 25-JAN-96 P00001          10 BackOrder

SQL> delete from vw_sales_det where client_no= 'C005';

0 rows deleted.

SQL> select * from vw_sales_det;

CLIENT ORDER_ ORDER_DAT PRODUC QTY_ORDERED ORDER_STAT
----- ----- -----
C002    019002 25-JAN-96 P00001          10 BackOrder

SQL> drop view vw_sales_det;

View dropped.
```

# ASSIGNMENT-7

## 1. Write a PL/SQL code for finding factorial of a given number

PL/SQL commands:

```
set serveroutput on
declare
    n number;
    i number;
    f number:=1;
begin
    n:=&x;
    for i in 1..n
        loop
            f:=f*i;
        end loop;
    dbms_output.put_line('Factorial of '||n||' is '|| f);
end;
/

```

Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_1.sql
Enter value for x: 5
old   6:      n:=&x;
new   6:      n:=5;
Factorial of 5 is 120
PL/SQL procedure successfully completed.
```

## 2. Write a PL/SQL code for calculating finding the sum of N numbers.

PL/SQL commands:

```
set serveroutput on
declare
    n number;
    i number;
    s number:=0;
begin
    n:=&n;
    for i in 1..n
        loop

```

```
s:=s+i;  
end loop;  
dbms_output.put_line('Sum of first '||n||' numbers is '|| s);  
end;  
/  
Output:
```

```
SQL> @E:\CSE_30\Ass7\asg_7_2.sql  
Enter value for n: 10  
old   6:      n:=&n;  
new   6:      n:=10;  
Sum of first 10 numbers is 55  
  
PL/SQL procedure successfully completed.
```

### 3. Write a PL/SQL code for finds a given year is leap year or not.

#### PL/SQL commands:

```
set serveroutput on  
declare  
    y number;  
begin  
    y:=&x;  
    if(mod(y,400)=0)then  
        dbms_output.put_line('Leap Year');  
    elsif((mod(y,4)=0) and (mod(y,100)!=0))then  
        dbms_output.put_line('Leap Year');  
    else  
        dbms_output.put_line('NOT a Leap Year');  
    end if;  
end;  
/
```

#### Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_3.sql  
Enter value for x: 2004  
old   4:      y:=&x;  
new   4:      y:=2004;  
Leap Year  
  
PL/SQL procedure successfully completed.
```

### 4. Write a PL/SQL code for finding maximum of three numbers. (Input will be given by the user).

#### PL/SQL commands:

```
set serveroutput on  
declare  
    a number;  
    b number;
```

```

c number;

begin

    a:=&a;

    b:=&b;

    c:=&c;

    if(a>b and a>c)then

        dbms_output.put_line(a || ' is the maximum');

    elsif(b>c)then

        dbms_output.put_line(b || ' is the maximum');

    else

        dbms_output.put_line(c || ' is the maximum');

    end if;

end;
/

```

Output:

```

SQL> @E:\CSE_30\Ass7\asg_7_4.sql
Enter value for a: 7
old   6:    a:=&a;
new   6:    a:=7;
Enter value for b: 21
old   7:    b:=&b;
new   7:    b:=21;
Enter value for c: 10
old   8:    c:=&c;
new   8:    c:=10;
21 is the maximum

PL/SQL procedure successfully completed.

```

**5. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 6 to 10. Store the radius and corresponding values of calculated area in an empty table named Areas, Consisting of two columns Radius and Area.**

PL/SQL commands:

```

set serveroutput on

drop table Areas;

create table Areas(radius number(5,3), area number(10,3));

declare

    r number;

    pi constant number(8,2):=3.14;

    area number(10,2);

begin

```

```

for r in 6..10
loop
    area:=pi*power(r,2);
    insert into Areas values(r,area);
end loop;
end;
/

```

Output:

```

SQL> @E:\CSE_30\Ass7\asg_7_5.sql

Table dropped.

Table created.

PL/SQL procedure successfully completed.

SQL> select * from Areas;

  RADIUS      AREA
  -----  -----
       6      113.04
       7      153.86
       8      200.96
       9      254.34
      10      314

```

6. Write a PL/SQL code block that will accept a client\_no from the user and adds the amount of Rs. 1000 to bal\_due column, has a minimum balance of Rs. 6000. The process is fire on client\_master.

PL/SQL commands:

```

set serveroutput on
declare
    cli_no varchar2(6):= '&client_no';
    t_c_no number(10,2);
begin
    select balance_due into t_c_no from client_master where client_no=cli_no;
    if(t_c_no>=6000)then
        t_c_no := t_c_no + 1000;
        update client_master set balance_due=t_c_no where client_no=cli_no;
    else
        dbms_output.put_line('The balance is below 6000.');
    end if;
end;
/

```

```
end if;
```

```
end;
```

```
/
```

Output:

```
SQL> @E:\CSE_30\Ass7\asg_7_6.sql
Enter value for client_no: C001
old  2:    cli_no varchar2(6):= '&client_no';
new  2:    cli_no varchar2(6):= 'C001';

PL/SQL procedure successfully completed.

SQL> set linesize 500;
SQL> select client_no, balance_due from client_master;

CLIENT  BALANCE_DUE
----- -----
C008      8000
C009      20000
C001      16000
C002          0
C003      5000
C004          0
C005      3000
C006          0

8 rows selected.
```

# **ASSIGNMENT-8**

- 1. a) Create a table whose structure will be as follows:**

**Table Name: Prime\_Entry**

Column Name	Data Type	Attributes
Num_id	Number(3)	Primary Key
Prime_num	Number(3)	Not Null

**COMMANDS:**

```
set serveroutput on;
create table prime_entry(
    num_id number(3) primary key,
    prime_num number(3) not null
);
```

```
create sequence seq
start with 1
increment by 1
/
```

**OUTPUT:**

```
SQL> @D:\SQL\DBMS\Ass8\asg8_1a.sql
Table created.

Sequence created.
```

- b) Write a PL/SQL block of code that will take a number from user and test whether the number is prime or not. If the number is prime, then enter into above table by generating NUMID automatically.**

**COMMANDS:**

```
set serveroutput on;
```

```
declare
    num number;
    j number;
    n number;
    i number;
    flag number;
    g number;
```

```
begin
```

```

num:=&n;
n:=TRUNC(num/2);
for i in 2..n
loop
    if(mod(num,i)=0)then
        flag:=1;
        exit;
    else
        flag:=0;
    end if;
end loop;
dbms_output.put_line('-----');
if(flag=1)then
    dbms_output.put_line(num||' is not prime');
else
    select seq.nextval into g from dual;
    insert into prime_entry values(g,num);
end if;
end;
/

```

### **OUTPUT:**

```

SQL> @D:\SQL\DBMS\Ass8\asg8_1b.sql
Enter value for n: 4
old 10:      num:=&n;
new 10:      num:=4;
-----
4 is not prime

PL/SQL procedure successfully completed.

SQL> @D:\SQL\DBMS\Ass8\asg8_1b.sql
Enter value for n: 5
old 10:      num:=&n;
new 10:      num:=5;
-----
PL/SQL procedure successfully completed.

SQL> @D:\SQL\DBMS\Ass8\asg8_1b.sql
Enter value for n: 7
old 10:      num:=&n;
new 10:      num:=7;
-----
PL/SQL procedure successfully completed.

SQL> select * from prime_entry;

  NUM_ID  PRIME_NUM
  -----  -----
        1          5
        2          7

```

c) Now add a checking for same prime number entry. It will show - 'Number already exists in database' for same prime number entry. Write a function to test whether given number exist or not.

**COMMANDS:**

set serveroutput on;

```
create or replace function prime_test(id number) return number
is
num number(20);
begin
    select num_id into num from prime_entry where prime_num=id;
    return 1;
exception
    when no_data_found then
        return 0;
```

end;

/

declare

num number;

j number;

i number;

n number;

flag number;

x number;

begin

num:=&n;

n:=TRUNC(num/2);

for i in 2..n

loop

if(mod(num,i)=0)then

flag:=1;

exit;

else

flag:=0;

end if;

end loop;

dbms\_output.put\_line('-----');

if(flag=1)then

dbms\_output.put\_line(num||' is not prime');

else

x:=prime\_test(num);

```

if(x=0)then
    insert into prime_entry values(seq.nextval, num);
else
    dbms_output.put_line('Already exists in the table. ');
end if;
end if;
end;
/

```

**OUTPUT:**

```

SQL> @D:\SQL\DBMS\Ass8\asg8_1c.sql

Function created.

Enter value for n: 4
old 9:      num:=&n;
new 9:      num:=4;
-----
4 is not prime

PL/SQL procedure successfully completed.

SQL> @D:\SQL\DBMS\Ass8\asg8_1c.sql

Function created.

Enter value for n: 7
old 9:      num:=&n;
new 9:      num:=7;
-----
Already exists in the table.

PL/SQL procedure successfully completed.

SQL> @D:\SQL\DBMS\Ass8\asg8_1c.sql

Function created.

Enter value for n: 11
old 9:      num:=&n;
new 9:      num:=11;
-----
PL/SQL procedure successfully completed.

SQL> select * from prime_entry;

  NUM_ID  PRIME_NUM
-----  -----
      1          5
      2          7
      3         11

```

**2. Create the following table:**

**Table Name: Acc\_details**

Column_Name	Data type	Size	Attributes
Acc_no	Varchar2	8	Primary Key
Name	Varchar2	20	Not Null
Address	Varchar2	20	Not Null
DOB	Date		Not Null
Sex	Char	1	Not Null, Values ('M', 'F')
Contact_no	Number	10	Not Null
Last_trans_date	Date		Not Null
Total_amt	Number	12,4	Not Null
Acc_status	Char	1	Not Null, Values ('A', 'I')

**Table Name: Transactions\_Acc**

Column_Name	Data type	Size	Attributes
Transaction_id	Number	8	Primary Key
Acc_no	Number	8	References Acc_details.Acc_no
Deposit_amt	Number	12,4	
Withdraw_amt	Number	12,4	
Mode_trans	Char	5	Not Null
Cheque_no	Number	6	Default 0
Trans_date	Date		Not Null

**When a specific account will be deleted then all the transaction details from Transactions\_Acc will be deleted for that account number.**

**COMMANDS:**

```
create table Acc_details(
    Acc_No varchar2(8) primary key,
    Name varchar2(20) not null,
    Address varchar2(50) not null,
    DOB date not null,
    sex char(1) check (sex in ('M', 'F')),
    contact_no number(10) not null,
    last_trans_date date not null,
    Total_cost number(14,2) not null,
    Acc_status char(1) not null check(Acc_status in ('A', 'I'))
);
```

```
create table Transaction_Acc(
    Transaction_Id number(8) primary key,
    Acc_No varchar2(8) references Acc_details on DELETE CASCADE,
    Deposit_amt number(12,4),
```

```
Withdraw_amt number(12,4),
Mode_trans char(5) not null,
Check_no number(6) default 0,
Trans_date date not null
);
```

```
desc Acc_details;
desc Transaction_acc;
```

```
insert into Acc_details values('001', 'AMIT', 'BK-256', '12-JAN-2012', 'M', 9836773258,
'13-JUN-2012', 12000, 'A');
```

```
select * from Acc_details;
```

```
insert into Transaction_Acc values(002, '001', 11000, 5000, 'A', 101, '12-JUN-2012');
insert into Transaction_Acc values(003, '001', 12000, 6000, 'B', 102, '13-JUL-2012');
```

```
select * from Transaction_Acc;
```

```
delete from Acc_details where Acc_no='001';
```

```
select * from Acc_details;
```

```
select * from Transaction_Acc;
```

## OUTPUT:

```
SQL> @D:\SQL\DBMS\Ass8\asg8_2.sql
Table created.

Table created.

Name          Null?    Type
-----
ACC_NO        NOT NULL VARCHAR2(8)
NAME          NOT NULL VARCHAR2(20)
ADDRESS        NOT NULL VARCHAR2(50)
DOB           NOT NULL DATE
SEX            CHAR(1)
CONTACT_NO    NOT NULL NUMBER(10)
LAST_TRANS_DATE NOT NULL DATE
TOTAL_COST    NOT NULL NUMBER(14,2)
ACC_STATUS    NOT NULL CHAR(1)

Name          Null?    Type
-----
TRANSACTION_ID NOT NULL NUMBER(8)
ACC_NO         NOT NULL NUMBER(8)
DEPOSIT_AMT   VARCHAR2(8)
WITHDRAW_AMT  NUMBER(12,4)
MODE_TRANS    NUMBER(12,4)
CHECK_NO       NOT NULL CHAR(5)
NUMBER(6)
TRANS_DATE    NOT NULL DATE
```

```
1 row created.
```

ACC_NO	NAME	ADDRESS	DOB	S	CONTACT_NO	LAST_TRAN	TOTAL_COST	A
001	AMIT	BK-256	12-JAN-12	M	9836773258	13-JUN-12	12000	A

```
1 row created.
```

```
1 row created.
```

TRANSACTION_ID	ACC_NO	DEPOSIT_AMT	WITHDRAW_AMT	MODE_	CHECK_NO	TRANS_DAT
2	001	11000	5000	A	101	12-JUN-12
3	001	12000	6000	B	102	12-JUL-12

```
SQL> delete from Acc_details where Acc_no='001';
```

```
1 row deleted.
```

```
SQL> select * from Acc_details;
```

```
no rows selected
```

```
SQL> select * from Transaction_Acc;
```

```
no rows selected
```

# ASSIGNMENT-9

**Problem Statement:** Deploy a project from GitHub to EC2.

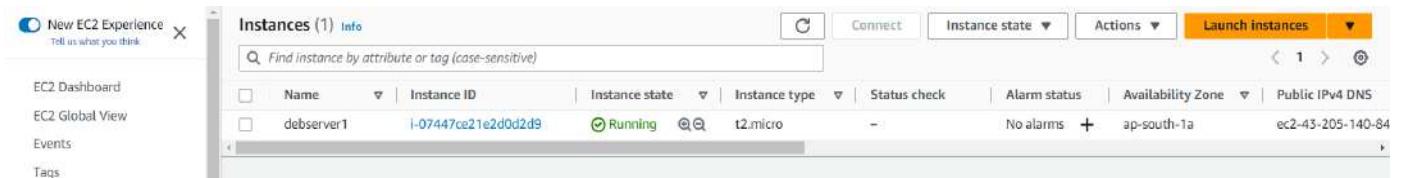
**Procedure:**

1. Go to GitHub Website <https://github.com/> and Sign In to your account.

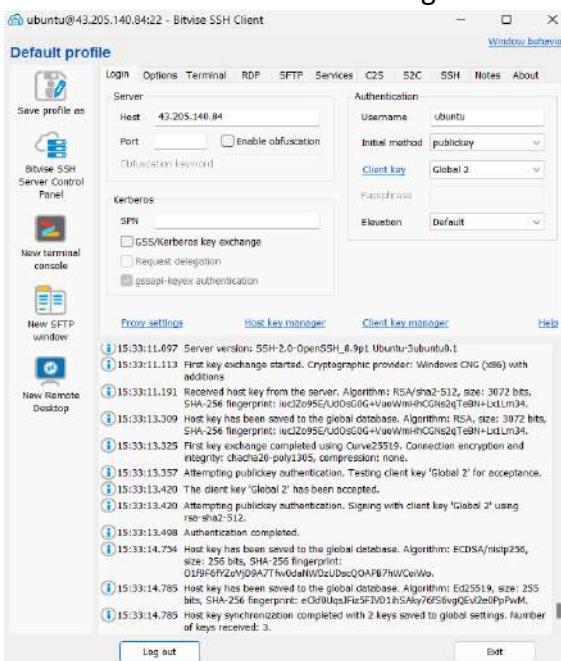


2. Also, Sign-In to your AWS account.

3. Create an EC2 instance (Refer to Ass7)



4. Connect the to the instance using the Bitvise SSH Client (Refer to Ass7)



5. Now Click on New Terminal Console option in the Left Sidebar of the Bitvise Client.

6. A terminal window will open and in it type the following commands:-

→ **sudo apt-get update & sudo apt-get upgrade**

(After few steps of progression, in case of any prompts asking (Y/N) press 'y' button and then press 'Enter' to continue. At the last stages if a UI appears on the screen, just press 'Enter' to continue. After the whole process is complete enter the next command as mentioned below)

→ **sudo apt-get install nginx**

(After few steps of progression, in case of any prompts asking (Y/N) press 'y' button and then press 'Enter' to continue. At the last stages if a UI appears on the screen, just press 'Enter' to continue. After the whole process is complete enter the next command as mentioned below)

→ **nginx -v**

```
ubuntu@ip-172-31-38-127:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```

(This command displays the nginx version installed in the server system)

→ **curl -Sf https://deb.nodesource.com/setup\_18.x | sudo -E bash -**

(This command downloads NodeJS files with all dependencies in our server system)

→ **sudo apt install nodejs**

(Press ‘Enter’ to continue when any UI appears on screen)

(This command installs NodeJS in our server system)

→ **node -v**

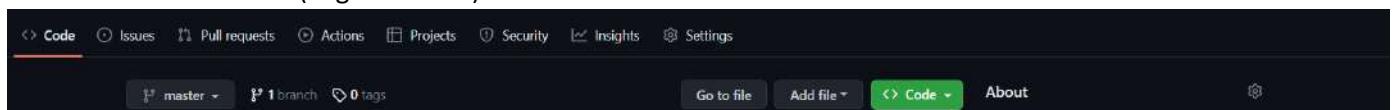
```
ubuntu@ip-172-31-38-127:~$ node -v  
v18.15.0
```

(This command displays the version of NodeJS installed in our server system)

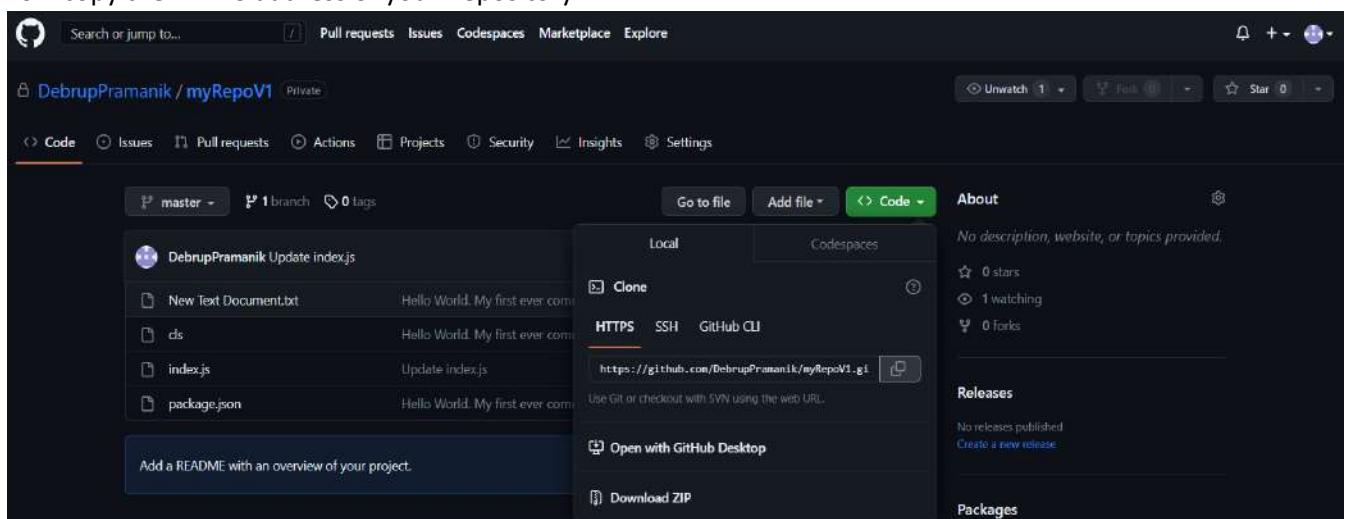
Now, minimize the terminal window. Go to the browser where our GitHub is Logged-In.

7. Go to your GitHub Repository which you want to upload in your EC2 server.

8. Click on the code button (in green color).



9. Now copy the HTTPS address of your Repository



10. Now return to the minimized terminal window and enter the following commands:-

→ **git clone https-address-you-just-copied-in-step-10**

```
ubuntu@ip-172-31-38-127:~$ git clone https://github.com/DebrupPramanik/myRepoV1.git  
Cloning into 'myRepoV1'...  
Username for 'https://github.com':
```

(Remember to paste your own https address that you copied in the above command in place of the one given in the screenshot)

(As shown in the screenshot, you will be asked to enter your username for GitHub. So mention your username there.)

After that you will be requested to provide your password. However, you have to enter your Account Token you generated instead of your password. If you don't have a Account Token then refer to Ass7 and create one for your GitHub account. Now copy-paste the Account Token (from the text file you have saved it) where it wants to mention your password. For pasting just Right click for a single time on the terminal where you want to paste (Note you won't be able to see your pasted token on the terminal as it is hidden by default. So just press ‘Enter’ to continue)

```
ubuntu@ip-172-31-38-127:~$ git clone https://github.com/DebrupPramanik/myRepoV1.git  
Cloning into 'myRepoV1'...  
Username for 'https://github.com': DebrupPramanik  
Password for 'https://DebrupPramanik@github.com':  
remote: Enumerating objects: 9, done.  
remote: Counting objects: 100% (9/9), done.  
remote: Compressing objects: 100% (8/8), done.  
remote: Total 9 (delta 2), reused 4 (delta 0), pack-reused 0  
Receiving objects: 100% (9/9), done.  
Resolving deltas: 100% (2/2), done.
```

→ **dir**

```
ubuntu@ip-172-31-38-127:~$ dir  
myRepoV1
```

(As seen this is the name of our cloned repository. This means a new directory has been created in our present working directory which has been named automatically to match the name of our Repository.)

→ cd myRepoV1/

```
ubuntu@ip-172-31-38-127:~$ dir  
myRepoV1  
ubuntu@ip-172-31-38-127:~$ cd myRepoV1/  
ubuntu@ip-172-31-38-127:~/myRepoV1$ █
```

(Now we enter into the directory)

→ ls -A

```
ubuntu@ip-172-31-38-127:~/myRepoV1$ ls -A  
.git 'New Text Document.txt' cls index.js package.json  
ubuntu@ip-172-31-38-127:~/myRepoV1$ █
```

(This command displays all the files in the current directory)

(We observe that we have all the files that we have in our Repository has been cloned in our directory in the server system)

→ npm install

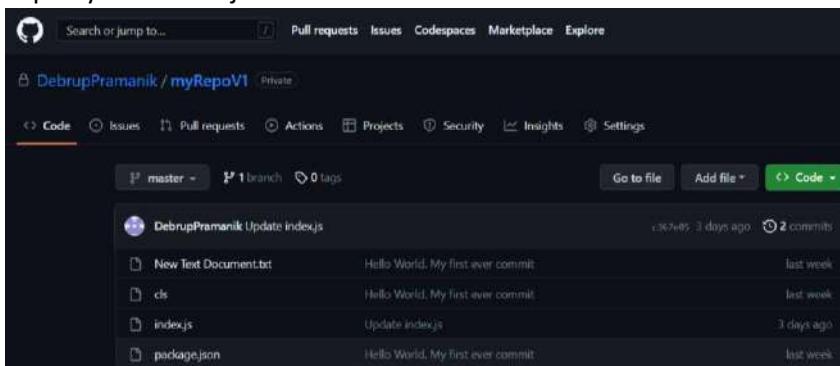
```
ubuntu@ip-172-31-38-127:~/myRepoV1$ npm install  
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.  
  
added 258 packages, and audited 259 packages in 10s  
  
18 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
npm notice  
npm notice New minor version of npm available! 9.5.0 -> 9.6.4  
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.6.4  
npm notice Run npm install -g npm@9.6.4 to update!  
npm notice  
ubuntu@ip-172-31-38-127:~/myRepoV1$ █
```

(This command installs the npm package manager)

Now before proceeding further we need to return back to GitHub. Minimize the terminal for now.

11. Go back to your Repository in Github.

12. Open your “index.js” file



13. Check the port no. specified in the program. It is specified in the app.listen() method as the first parameter. Here it is ‘4000’. Copy or remember this no. as it is the port no. and will be required to connect to our website.

```
11 lines (9 sloc) | 205 Bytes  
  
1 const express = require('express')  
2 const app = express()  
3  
4 app.get('/', function (req, res) {  
5   res.send('Hello. My Name is Spider-Man!!!')  
6 })  
7  
8 app.listen(4000, ()=>{  
9   console.log("Started server");  
10 }  
11 )
```

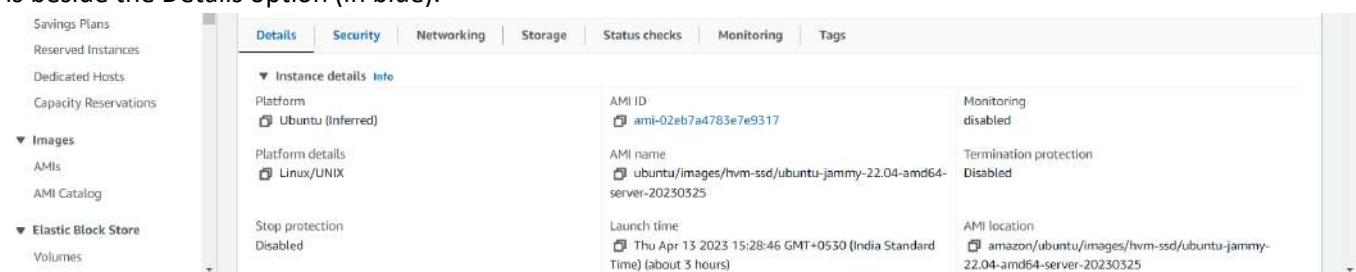
We have to add this port no. to our EC2 instance security group rule otherwise we won't be able to access the website from anywhere.

14. Now go back to your AWS EC2 instances page.

15. Click on the instance that is being used.



16. Scroll down until you find a section bar where by default the details option is selected. Select the Security option. It is beside the Details option (in blue).



Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-09b78026b8d8185ec	443	TCP	0.0.0.0/0	launch-wizard-5
-	sgr-0a5fb8fb952398167	80	TCP	0.0.0.0/0	launch-wizard-5
-	sgr-083c26f7314e7b3ac	22	TCP	0.0.0.0/0	launch-wizard-5

17. Then click on the security groups link under security groups



Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-09b78026b8d8185ec	443	TCP	0.0.0.0/0	launch-wizard-5
-	sgr-0a5fb8fb952398167	80	TCP	0.0.0.0/0	launch-wizard-5
-	sgr-083c26f7314e7b3ac	22	TCP	0.0.0.0/0	launch-wizard-5

18. Then click on Edit Inbound Rules button.

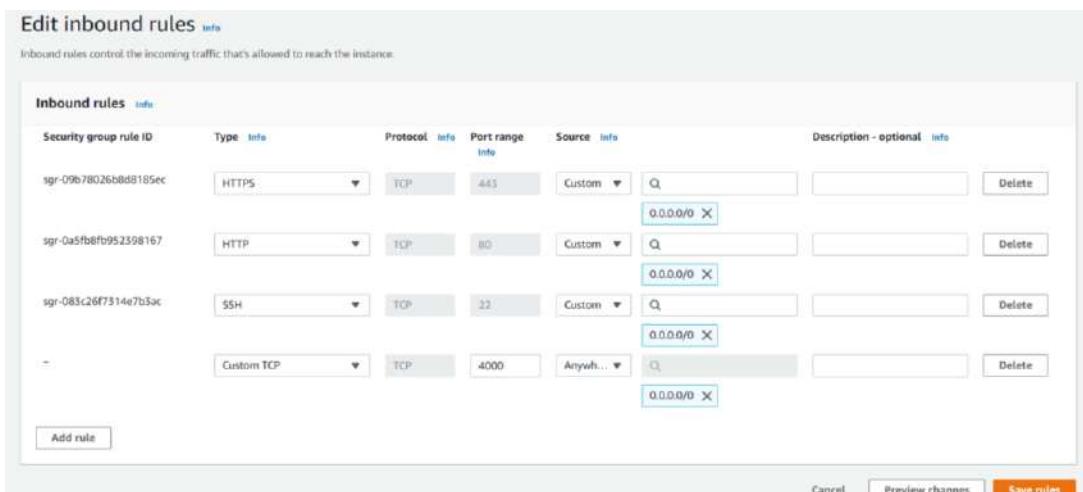


19. Click on the Add Rule button.



20. A new Row will be generated. Let the type remain Custom TCP. Under Port Range write your Port no. you want to open. In this case we have 4000 port no. as we found out earlier in our index.js code. Next in source click on the search box and the first option with value 0.0.0.0/0 should be selected.

21. Now click on the save rules button.



We have successfully added the Port No. to our Inbound rules. Now we can access our website. But first we need to start our server.

22. We return back to the terminal and type:-

→ node index.js

```
ubuntu@ip-172-31-38-127:~/myRepoV1$ node index.js
Started server
```

Our server has started and it is also reflected by the terminal prompt. Now to check we need to open another browser and type in the IPv4 address of our EC2 server to access our website.

23. Now copy the IPv4 address of your EC2 server and paste it in another browser. But before pressing Enter add a colon (:) and then mention the Port No. mentioned in the index.js file. For our case it is 4000. In our case the full address resulted in the one in the below screenshot.

```
43.205.140.84:4000
```

24. Now press Enter to load the website.



25. We have successfully deployed our project from GitHub to our EC2 server.

Now if we want to change something in our project or file or code we will follow these general steps:

26. Suppose we want to modify the displayed message. So open the index.js file in GitHub.  
27. After opening we click on the pen icon on the right corner side of the code viewer.

```
11 lines (9 sloc) 205 Bytes
1 const express = require('express')
2 const app = express()
3
4 app.get('/', function (req, res) {
5   res.send('Hello. My Name is Spider-Man!!!')
6 })
7
8 app.listen(4000, ()=>{
9   console.log("Started server")
10 })
11 )
```

28. We then modify the string passed through the res.send() method.

```
11 lines (9 sloc) 225 Bytes
1 const express = require('express')
2 const app = express()
3
4 app.get('/', function (req, res) {
5   res.send('Hello. My Name is Spider-Man!!! Nice to meet You!!!')
6 })
7
8 app.listen(4000, ()=>{
9   console.log("Started server")
10 })
11 )
```

29. Now after editing, scroll-down and click the Commit changes button.

30. Our changes have finally been committed in our Repository in GitHub.

However our changes have not been reflected in our Remote Server. For that we have to ‘pull’ the new updated files into the repository directory in our Remote Server.

31. We have to Close our already running Server. For this, go to the Terminal which we have been working with. Then press <CTRL + C> shortcut to close the server.

```
Started server
^C
```

```
ubuntu@ip-172-31-38-127:~/myRepoV1$
```

Our server has been stopped.

32. Now type:-

→ git pull

(Enter the username when asked)

(Enter your account Token as your Password when asked for password)

(Right click once to paste, then press Enter)

```
ubuntu@ip-172-31-38-127:~/myRepoV1$ git pull
Username for 'https://github.com': DebrupPramanik
Password for 'https://DebrupPramanik@github.com':
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 685 bytes | 228.00 KiB/s, done.
From https://github.com/DebrupPramanik/myRepoV1
  c367e05..188da63  master      -> origin/master
Updating c367e05..188da63
Fast-forward
 index.js | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)
ubuntu@ip-172-31-38-127:~/myRepoV1$
```

Now we have to restart the server.

→ **node index.js**

(We restarted the server)

33. We now have to Refresh our browser where we have our website open.



The changes have been successfully reflected. This is how we have to edit and update our project if required.

We have successfully completed our task of Deploying our project from GitHub to our EC2 server.

# ASSIGNMENT – 10

**Problem Statement:** Deploy project from GitHub to EC2 by creating new security group and user data.

## Procedure:

1. Sign in to your AWS account.
2. Go to your EC2 dashboard
3. Scroll down and Click on Security Groups option on the left side nav bar under Network & Security option.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with options like Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Key Pairs. The main area is titled 'Resources' and displays various Amazon EC2 resources in the Asia Pacific (Mumbai) Region. It includes sections for Instances (running), Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. A callout box at the bottom right of the main area provides information about easily sizing, configuring, and deploying Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server.

4. Select all the Security Groups other than the one named “default”.

The screenshot shows the AWS Security Groups list. There are two entries: 'default' (Security group ID: sg-0e4361e5c76c8c036, VPC ID: vpc-0a33deec3fd6dc096, Owner: 728364961341) and 'mysecgr1' (Security group ID: sg-0abe126418b4ea2a8, VPC ID: vpc-0a33deec3fd6dc096, Owner: 728364961341). The 'mysecgr1' row is selected, indicated by a blue border around its columns.

5. Then Click on the Actions button.

The screenshot shows the AWS Security Groups list with the Actions button dropdown open. The options listed are: Manage tags, Manage stale rules, Copy to new security group, and Delete security groups. The 'Delete security groups' option is highlighted with a blue border.

6. Scroll-Down the dropdown list until you find the “delete all security groups” option. Click on it.

The screenshot shows the AWS Security Groups list with the Actions button dropdown open. The 'Delete security groups' option is selected, indicated by a blue border around its text.

7. Now only the “default” security group remains and we keep it that way.

The screenshot shows the AWS Security Groups list. Only the 'default' security group (Security group ID: sg-0e4361e5c76c8c036, VPC ID: vpc-0a33deec3fd6dc096, Owner: 728364961341) remains in the list, while 'mysecgr1' has been removed.

8. Now click on the “Create Security Group” button.

The screenshot shows the AWS Security Groups list with the 'Create security group' button highlighted in orange at the top right of the page.

9. Now start by giving a name to the security group and giving its description (anything).

Let the VPC remain unchanged.

The screenshot shows the 'Create security group' page. In the 'Basic details' section, the 'Security group name' is set to 'mysec1' and the 'Description' is 'mysec1'. A dropdown for 'VPC' shows 'vpc-0a33deec5fd6dc096'.

10. Next, we will add Inbound Rules. Start adding by clicking the Add rule button. These include:

a) SSH

The screenshot shows an 'Inbound rules' table with one row. The 'Type' is 'SSH', 'Protocol' is 'TCP', 'Port range' is '22', 'Source' is 'Anywh...', and 'Description' is '0.0.0.0/0'.

b) HTTP

The screenshot shows an 'Inbound rules' table with one row. The 'Type' is 'HTTP', 'Protocol' is 'TCP', 'Port range' is '80', 'Source' is 'Anywh...', and 'Description' is '0.0.0.0/0'.

c) HTTPS

The screenshot shows an 'Inbound rules' table with one row. The 'Type' is 'HTTPS', 'Protocol' is 'TCP', 'Port range' is '443', 'Source' is 'Anywh...', and 'Description' is '0.0.0.0/0'.

d) Custom TCP

The screenshot shows an 'Inbound rules' table with one row. The 'Type' is 'Custom TCP', 'Protocol' is 'TCP', 'Port range' is '4000', 'Source' is 'Anywh...', and 'Description' is '0.0.0.0/0'.

The last one with custom TCP has a specific port range that we require to connect to our project. It has been specified in our index.js file (refer Ass9).

Now the final Inbound Rules section should look like this.

The screenshot shows the 'Inbound rules' table with four rows, each corresponding to the rules defined above: SSH (port 22), HTTP (port 80), HTTPS (port 443), and Custom TCP (port 4000). Each row has a 'Delete' button.

11. Next outbound rules and all other sections remain unchanged. Now Click on the create security group button.

The screenshot shows the 'Outbound rules' section. It has a table with one row where 'All traffic' is selected. Below the table is a 'Tags - optional' section with a note about tags and a 'Create security group' button at the bottom right.

12. Now go back to the security groups list and click on the security group ID of the newly created Security Group.

Name	Security group ID	Description	VPC ID
mysec1	sg-0493398d43b761e55	mysec1	vpc-0a33dec3fd6dc096
default	sg-0e4361e5c76c8c036		

Inbound rules (4)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0677b32c36bf02194	IPv4	HTTPS	TCP	443
-	sgr-02f914343500960...	IPv4	SSH	TCP	22
-	sgr-0ff95fa0debaeccc64	IPv4	HTTP	TCP	80
-	sgr-0d92a3e25bf3ad37	IPv4	Custom TCP	TCP	4000

After clicking we can view the inbound rules that we added during its creation.

13. Now we go to the instances section from the left side nav bar.

14. Now we Create a new EC2 instance. Click on the Launch Instance button.

Instances Info

Find instance by attribute or tag (case-sensitive)

Actions ▾ Launch instances ▾

No instances  
You do not have any instances in this region

Now,

a) Give the name

Name and tags

Name: definst1

Add additional tags

b) Select Ubuntu as OS.

Application and OS Images (Amazon Machine Image)

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

Quick Start

Amazon Linux	aws Lambda	AWS Mac	Ubuntu	Windows	Red Hat
--------------	------------	---------	--------	---------	---------

Amazon Machine Image (AMI)

c) Select a keypair or generate a new one if none is available.

Key pair (login)

Key pair name (required): diskkey2

Create new key pair

d) Then under Network settings select the Select Existing Security Group option.

Network settings

Network: vpc-0a33dec3fd6dc096

Subnet info: No preference (Default: subnet in any availability zone)

Auto-assign public IP

Enable

Firewall (security groups)

Select existing security group

Compare security group rules

- e) Now under the security groups dropdown menu select the one we just created.

It should look like this.....

- f) Now scroll down and click on the Advanced Details option.

- g) Now again scroll-down to the newly appeared sub-sections until you find User Data section.

- h) Write the following commands in the given box. Remember this user data is given to execute the given commands once the server starts. So essentially, we can provide all commands that we entered in our Assignment 9 previously and execute them without connecting to our server itself!! They will be executed sequentially.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
```

Now, here is a caveat. We have created a private repository in GitHub. So, whenever we run the git clone command it asks for our username and password. Hence this cannot be executed directly through our User Data instructions. We have to connect manually and enter all commands starting from the git clone command.

- i) Now we click on the launch instance button.



## 15. Now we Click on the 'Instance Id' link of our newly created server in our Instances list.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	debserver1	i-0a6ab24417f81fffb	<span>Running</span>	t2.micro	<span>Initializing</span>	No alarms	ap-south-1a

## 16. Now click on the connect button

Instance summary for i-0a6ab24417f81fffb (debserver1)

Public IPv4 address: 3.110.154.34 | [open address](#)

Private IPv4 addresses: 172.31.41.246

Public IPv4 DNS: ec2-3-110-134-34.ap-south-1.compute.amazonaws.com | [open address](#)

## 17. Again, click on the connect button

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID: i-0a6ab24417f81fffb (debserver1)

Public IP address: 3.110.154.34

User name: ubuntu

Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel | Connect

## 18. After this anew Tab will open with a Bash Terminal that is of our remote EC2 server!

Here we can type all our required commands that we used to type in a similar terminal by connecting to our remote server through our Bitvise SSH client software in our previous assignments.

The terminal window shows a standard Ubuntu 22.04 LTS prompt: debserver1:~ root@i-0a6ab24417f81fffb:~%

The session title is "Connect to instance [EC2 Man...]" and the tab title is "EC2 Instance Connect".

## 19. Now type the following commands in the terminal:-

→ git clone <https://github.com/> ..... //Your GitHub Repository URL

Give your Username of GitHub when asked.

Give your account Token when your Password is asked.

```
ubuntu@ip-172-31-41-246:~$ git clone https://github.com/DebrupPramanik/myRepoV1.git
Cloning into 'myRepoV1'...
Username for 'https://github.com': DebrupPramanik
Password for 'https://DebrupPramanik@github.com':
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.
```

→ cd YourRepositoryname/

```
ubuntu@ip-172-31-41-246:~$ cd myRepoV1/
ubuntu@ip-172-31-41-246:~/myRepoV1$ █
```

→ npm install

```
ubuntu@ip-172-31-41-246:~/myRepoV1$ npm install
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. See https://v8.dev/blog/math-random for details.

added 258 packages, and audited 259 packages in 15s

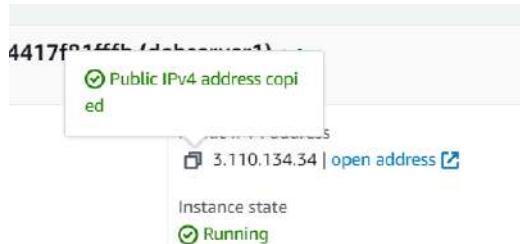
18 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 9.5.1 -> 9.6.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.6.5
npm notice Run npm install -g npm@9.6.5 to update!
npm notice
```

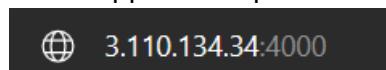
→ node index.js

```
ubuntu@ip-172-31-41-246:~/myRepoV1$ node index.js
Started server
```

20. Now copy and paste the Public IPv4 address of your EC2 instance in another browser.



21. Now append the port no. 4000 (for our case) to the IP address in the browser with a ":" sign.



We have successfully Deployed a project from GitHub to EC2 by creating a new Security group and User Data.

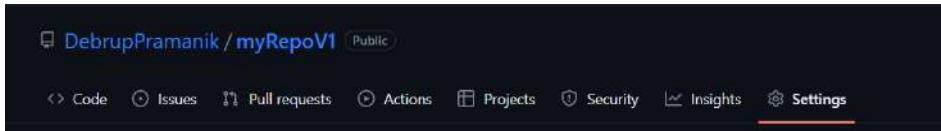
# ASSIGNMENT – 11

**Problem Statement:** Build Scaling plans in AWS that balance load on different EC2 instances.

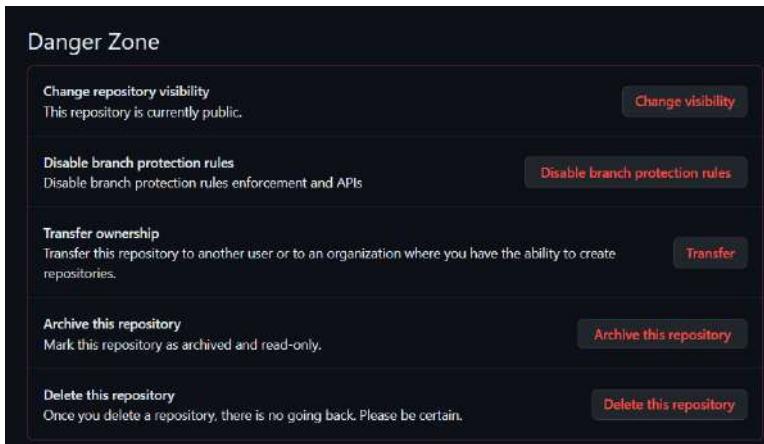
## Procedure:

1. Sign-in to GitHub. Make sure the Repository which will be cloned is made public or not.

a. For This, select the “settings” tab of your repository.



b. Next Scroll Down until you reach the danger-zone section.



c. In here, check The Change Repository visibility option. Here, we can see the repository is currently Public.

If it shows Private then click on the Change Visibility option and follow the on-screen Instructions to make the repository Public.

2. Now in another tab open AWS and Sign-in to your console.

3. Now go to your EC2 dashboard.

4. From the Left Side Nav-bar Go to your Instances Section. Under it Click the launch templates button.

A screenshot of the AWS EC2 Dashboard. On the left, there's a navigation bar with links like "New EC2 Experience", "EC2 Dashboard", "EC2 Global View", "Events", "Tags", "Limits", "Instances", "Instances", "Instance Types", "Launch Templates", "Spot Requests", "Savings Plans", "Reserved Instances", "Dedicated Hosts", and "Capacity Reservations". The main area shows "Resources" with a table of counts for various EC2 components. Below that are sections for "Launch instance" (with a "Launch instance" button) and "Service health" (showing the "Region: Asia Pacific (Mumbai)"). To the right, there's a sidebar for "Account attributes" and an "Explore AWS" section.

5. Now click on the Create Launch Templates button.

A screenshot of the "EC2 launch templates" page. The title is "Streamline, simplify and standardize instance launches". It includes a "New launch template" button. Below it, there are sections for "Benefits and features" (listing "Streamline provisioning", "Simplify permissions", and "Documentation"), and "Streamline provisioning" (with a "Learn more" link). There's also a "Get 1 in 40% Better Price Performance" callout.

6. Now, give a name and description for your EC2 template you are about to create. Here we gave the same for both the fields.

Next, Check the “Provide Guidance” box.

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name - required

debtemp1

Must be unique to this account. Max 128 chars. No spaces or special characters like %, \*, @, #.

Template version description

debtemp1

Max 256 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Template tags

▶ Source template

7. Next, under the quick start menu select Ubuntu as the OS.

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | S |

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-02eb7a47b3e7e9317 (64-bit (x86)) / ami-05dcff6fb7af5fc9 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture

AMI ID

64-bit (x86) | ami-02eb7a47b3e7e9317 | Verified provider

8. Under Instance type select t2.micro type of configuration.

▼ Instance type [Info](#)

Advanced

Instance type

t2.micro

Family: t2 - 1 vCPU - 1 GiB Memory Current generation: true

Free tier eligible

On-Demand Linux pricing: 0.0124 USD per Hour

On-Demand Windows pricing: 0.017 USD per Hour

On-Demand RHEL pricing: 0.0724 USD per Hour

On-Demand SUSE pricing: 0.0124 USD per Hour

All generations

Compare instance types

9. Select Existing Key-Pair and Security Group and if not applicable then Generate or Create a Key-Pair or Security Group wherever required.

▼ 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

debkey2

Create new key pair

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template

When you specify a subnet, a network interface is automatically added to your template.

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.

Select existing security group

Create security group

Security groups [Info](#)

Select security groups

mysec1 sg-0493398d43b761e55 X

VPC: vpc-0a33deec3fd6dc096

Compare security group rules

## 10. Now, Click on the Advanced Group Section at the bottom.

Scroll Down to User Data Section and paste the following commands in the provided box.

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone YourRepositoryURLhere
cd YourRepositoryNamehere/
npm install
node index.js
```

After adding the commands, it will look like this.....

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/DebrupPramanik/myRepoV1.git
cd myRepoV1/
npm install
node index.js
```

## 11. Now click on the Create Launch Template button.

Don't include in launch template

Metadata response hop limit: [Info](#)

Don't include in launch template

Allow tags in metadata: [Info](#)

Don't include in launch template

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/DebrupPramanik/myRepoV1.git
cd myRepoV1/
npm install
node index.js
```

User data has already been base64 encoded

▼ Summary

Software Image (AMI)  
Canonical, Ubuntu, 22.04 LTS, ... [read more](#)  
ami-02eb7a4785e7e9517

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
mysec1

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, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch templates (1) <small>info</small>				
Launch template ID	Launch template name	Default version	Latest version	Actions
lt-0d93dae5bfe8072d5	debtemp1	1	1	<a href="#">Create launch template</a>

12. Now from the Left side Nav Bar go to Auto Scaling Groups under Auto-Scaling section.

The screenshot shows the AWS Management Console with the 'Auto Scaling' section selected in the left navigation bar. The main content area displays the 'Launch templates' page, which lists a single entry: 'debtemp1' with a 'Launch template ID' of 'lt-0d93dae5bfe8072d5'. The navigation bar also includes tabs for 'Services' and 'Search', and shows the location 'EC2 > Launch templates'. The top right corner displays the user's name 'Mumbai' and the date 'CSE\_2020\_30'.

13. Now click on Create Auto Scaling Group.

This screenshot shows the first step of the 'Create Auto Scaling group' wizard. The title reads 'Amazon EC2 Auto Scaling helps maintain the availability of your applications'. Below the title is a description of what Auto Scaling groups are: 'Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.' A prominent orange button at the bottom right says 'Create Auto Scaling group'.

14. Now, Give a unique name to your new Auto Scaling Group. Also select the Launch Template that we recently created by using the drop-down menu under Launch Templates section.

This screenshot shows the second step of the 'Create Auto Scaling group' wizard. The 'Name' field is filled with 'deb\_asg1'. The 'Launch template' dropdown is set to 'debtemp1'. Below the dropdown is a link 'Create a launch template'. At the bottom, there is a 'Version' dropdown with 'Latest (1)' selected. The 'Switch to launch configuration' link is visible above the dropdown.

15. Under the selected Launch Template click on the version option and select Latest.

This screenshot shows a close-up of the 'Version' dropdown menu. The 'Latest (1)' option is highlighted with a blue selection bar. The dropdown also contains 'Default (1)' and 'Default (1)' again, with the number '1' below it.

**16.** Now click on the Next button.

**17.** After that, Under Availability Zones and Subnets select all the zones that appear.

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0a33deec3fd6dc096  
172.31.0.0/16 Default

Create a VPC

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

ap-south-1a | subnet-0bbe74a9835a07e58 X  
172.31.32.0/20 Default

ap-south-1b | subnet-0916d7caf57f5d661 X  
172.31.0.0/20 Default

ap-south-1c | subnet-09ddf920c63dcde50 X  
172.31.16.0/20 Default

Create a subnet

**18.** Again, click on the Next button.

**19.** Now Under Load Balancers select the Attach to a New Load balancer option.

Load balancing

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

**20.** Now select Internet-Facing under Load balancer scheme.

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

Internal

Internet-facing

**21.** Under Listeners and Routing enter the port no. of the project and select Create target group followed by giving the target group a name.

Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	4000	<input type="button" value="Create a target group"/> New target group name An instance target group with default settings will be created. debasg1

**22.** Now click on the next button.

**23.** After clicking on the Next button, a new page will open. Under Group Size mention:

- a) Desired Capacity = 2
- b) Minimum Capacity = 2
- c) Maximum Capacity = 3

Group size - optional

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

Minimum capacity

Maximum capacity

**24.** Now under Scaling policies Choose the Target Tracking Scaling policy option.

Select the metric type as Average CPU utilization.

Set Target Value to 50.

Set Warm-Up time to 300 seconds under Instances Need.

**Scaling policies - optional**

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

**Target tracking scaling policy**  
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

**None**

Scaling policy name

Metric type

Target value

Instances need  
 seconds warm up before including in metric

Disable scale in to create only a scale-out policy

**25.** Then click on next.

**26.** Nothing to do in Notifications page. So again, click on the Next button.

**27.** No tags needed. Again, click on the Next button.

**28.** Now Review your Auto-Scaling Group you are going to create. Now click on the Create Auto-Scaling Group button.



**29.** Now we can go to the Auto Scaling Groups section and find our newly created Auto Scaling Group.

Auto Scaling groups (1) <a href="#">Info</a>						
<input type="checkbox"/> Name		Launch template/configuration	Instances	Status	Desired capacity	Min
<input type="checkbox"/>	debashg1	debtemp1   Version Latest	2	-	2	2

**30.** Return to the Instances Page using the Left side Nav bar.

Name	Instance ID	Instance state	Instance type	Status check
-	i-0a7cf75ace548840f	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>
-	i-0c51d43594a967f2b	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>

**31.** Click on the First instance. Copy its public IPV4 DNS.

Instance summary for i-0a7cf75ace548840f <a href="#">Info</a>			
Updated less than a minute ago			
Instance ID	Public IPv4 address	Private IPv4 addresses	
i-0a7cf75ace548840f	<a href="#">13.126.127.63</a>   <a href="#">open address</a>	<a href="#">172.31.14.78</a>	
IPv6 address	Instance state	Public IPv4 DNS	
-	<span>Running</span>	<a href="#">ec2-13-126-127-63.ap-south-1.compute.amazonaws.com</a>   <a href="#">open address</a>	

32. Paste it in another browser.



## Welcome to nginx!

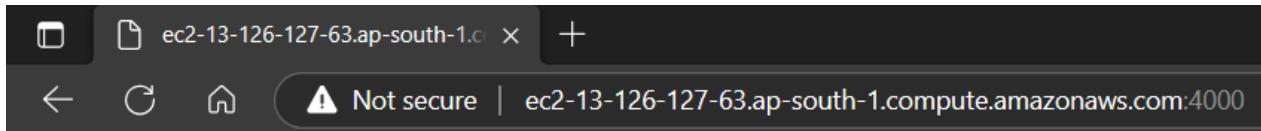
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

Thank you for using nginx.

We can successfully access the webpage.

33. Now to access our project webpage we need to append the port no. (4000) of our project with a ":"



Hello. My Name is Spider-Man!!! Nice to meet You!!!

34. To test our Auto-Scaling Group actually works we need to crash or overload the existing instance servers. Then only our Auto-Scaling Group will provide fresh instance servers automatically in case of crash; or it can provide extra servers to handle overloads.

35. We will now **CRASH THE SERVER INSTANCES** by terminating them.

36. Go to the instances page. Select the server instances.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-0a7cf75ace548840f	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1b	ec2-13-126-127-63
-	i-0c51d43594a967f2b	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1a	ec2-3-109-184-169

37. Now click on the Instance State button up top. From the drop-down select the Terminate instance option.

Instances (2/2) <a href="#">Info</a>		Connect	Instance state	Actions	Launch instances
<input type="checkbox"/>	Find instance by attribute or tag (case-sensitive)	<a href="#">Stop instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input checked="" type="checkbox"/>	Instance state = running	<a href="#">Start instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input checked="" type="checkbox"/>	Instance state = running	<a href="#">Reboot instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input checked="" type="checkbox"/>	i-0a7cf75ace548840f	<a href="#">Hibernate instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input checked="" type="checkbox"/>	i-0c51d43594a967f2b	<a href="#">Terminate instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>

38. Refresh the instances the page from time to time. After few seconds the instances will get terminated.

39. Wait for some time. Keep refreshing using the refresh button on the left side of Instance state button.

40. After some you will notice a new instance server will appear automatically! To help finding it more easily we need to activate the instance running filter. Click on the search box below the Instances Section Heading. Start typing running.

Select the option "Instance state = running option in the suggestion dropdown. The filter will be activated.

You can alternatively type the command directly in the search-box directly.

Now, only the running instances will be shown in the list.

Instances (1) <a href="#">Info</a>		Connect	Instance state	Instance type
<input type="checkbox"/>	running	<a href="#">Stop instance</a>		<a href="#">Actions</a>
<input type="checkbox"/>	Use: "running"	<a href="#">Start instance</a>		<a href="#">Actions</a>
<input type="checkbox"/>	filters	<a href="#">Reboot instance</a>		<a href="#">Actions</a>
<input type="checkbox"/>	API filters values	<a href="#">Hibernate instance</a>		<a href="#">Actions</a>
<input type="checkbox"/>	Instance state = running	<a href="#">Terminate instance</a>		<a href="#">Actions</a>

41. After some few seconds of refreshing we will be able to see two new servers are running.

Instances (2) <a href="#">Info</a>		Connect	Instance state	Actions	Launch instances
<input type="checkbox"/>	Find instance by attribute or tag (case-sensitive)	<a href="#">Stop instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input type="checkbox"/>	Instance state = running	<a href="#">Start instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>

Now again copy paste the new public IPv4 DNS of the first instance and port no. in the other browser to see if the instances are working. It will be working.

So, our Auto-Scaling Group can handle instance crashing by providing new fresh instances.

**42.** We will now **OVERLOAD THE SERVER INSTANCES** by running scripts and increasing CPU utilization value above the threshold that we specified during Configuration of the Auto scaling group.

**43.** For it we will use:

- a) Use Bitvise SSH client for instance 1.
- b) Use direct connect terminal in AWS for instance 2.

**44. For Instance-1:**

- a) Copy the public IPv4 address
- b) Open Bitvise SSH client.
- c) Paste the IP and select/specify the necessary options. (**Refer Ass7**)
- d) Now Log-In to your server.
- e) Open the new Terminal.
- f) Now enter the command:

→ **nano infil.sh**

- g) After the command a new nano Editor window will open. Type the following in it.

```
#!/bin/bash
while true
do
    echo "Loop running"
done
```

```
GNU nano 6.2
#!/bin/bash
while true
do
    echo "Loop running"
done
```

- h) Now, to save and close the shell script we need to press the following shortcuts and keys sequentially:

**Ctrl+X**

**Y**

**Enter**

- i) Now you will be returned back to the terminal.

- j) Now type the following commands:

→ **chmod +x infil.sh**

(Used to give the execute permission for the infil.sh file)

```
ubuntu@ip-172-31-13-76:~$ nano infil.sh
ubuntu@ip-172-31-13-76:~$ chmod +x infil.sh
```

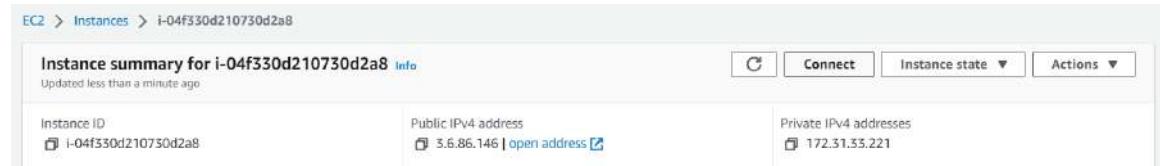
→ **./infil.sh**

(Used to execute the infil.sh script)

- k) Now the script will start running infinitely!
  - l) DO NOT CLOSE THE TERMINAL. Keep it minimized.

#### **45. For Instance-2:**

- a) Click on the instance 2.
  - b) Now click on the connect button



- c) Click on connect again
  - d) After the terminal gets opened we again follow the same steps as we did for instance-2
  - e) Enter the command:  
→ **nano infil.sh**
  - f) After the command a new nano Editor window will open. Type the following in it.

```
#!/bin/bash
while true
do
    echo "Loop running"
done
```

- g) Now, to save and close the shell script we need to press the following shortcuts and keys:  
**Ctrl+X**  
**Y**  
**Enter**
  - h) Now you will be returned back to the terminal.
  - i) Now type the following commands:

→ chmod +x infil.sh

(Used to give the execute permission for the infil.sh file)

→ ./infil.sh

(Used to execute the `infil.sh` script)

- j) Now the script will start running infinitely!
  - k) DO NOT CLOSE THE TERMINAL TAB. Go back to the previous tab to keep working in AWS.

**46.** Now go to the instances page.

**47.** Select both the instances.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
✓ -	i-0f5106bab83bb8ec1	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1b	ec2-43-204-214-47
✓ -	i-0naef30d210730d2a8	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1a	ec2-3-5-86-146.ap-south-1.amazonaws.com

48. Click on the instances white bar at the bottom of the page.

The screenshot shows the AWS CloudWatch Metrics Instances page. At the top, there are buttons for 'Instances (2/2)', 'Info', 'Connect', and 'Instance state'. Below is a search bar with placeholder text 'Find instance by attribute or tag (case-sensitive)'. A table lists two instances:

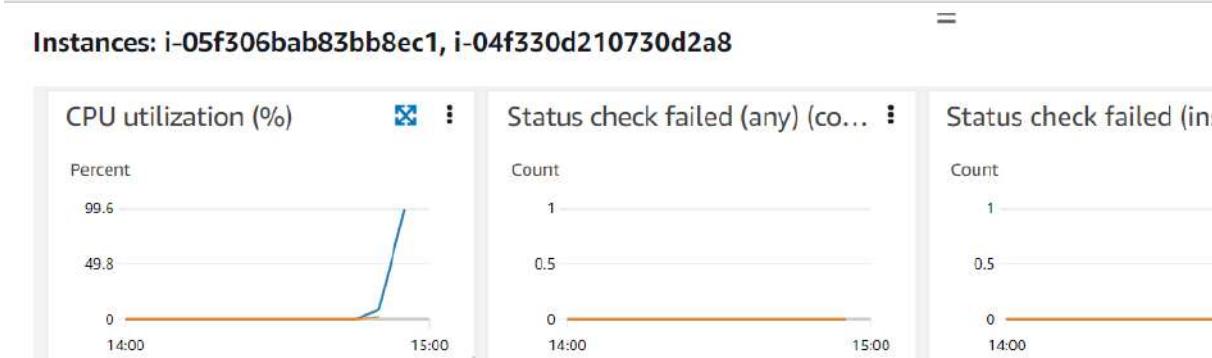
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
-	i-05f306bab83bb8ec1	Running	t2.micro	2/2 checks passed	No alarms
-	i-04f330d210730d2a8	Running	t2.micro	2/2 checks passed	No alarms

At the bottom of the page, a grey bar displays the text 'Instances: i-05f306bab83bb8ec1, i-04f330d210730d2a8'.

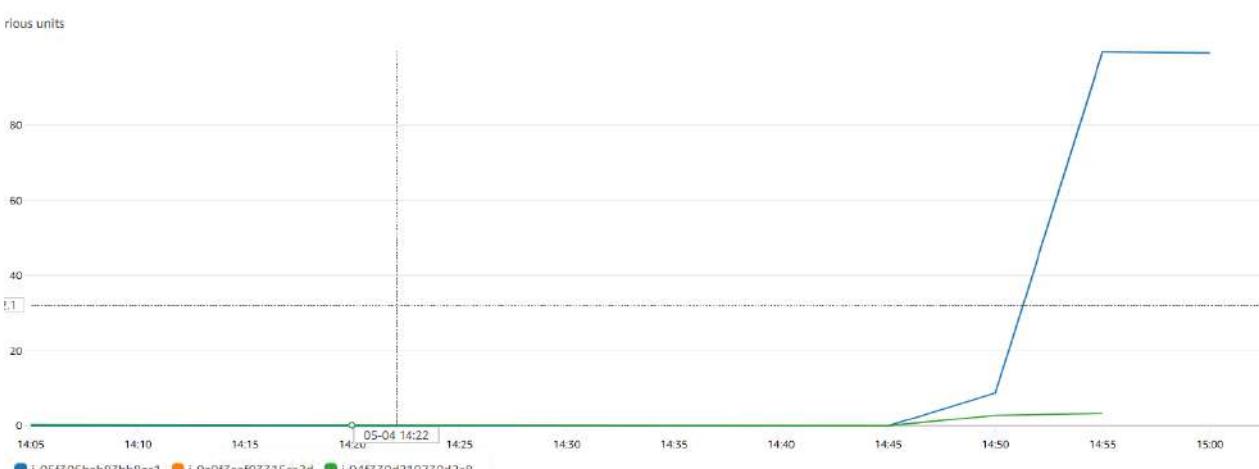
49. Now drag the two bars to expand the view.

The screenshot shows the same AWS CloudWatch Metrics Instances page, but the view has been expanded. The top bar now includes a maximize icon (two horizontal lines) and a close icon (an 'X').

50. We are interested only in the CPU utilization graph. Click on the maximize icon by hovering over the graph as shown in the fig to maximize this graph.



51. Our 1<sup>st</sup> instance has already crossed over 50% utilization. That's why we can see already a new third instance has been initiated by our auto-scaling group to compensate for the overload.



52. There can only be 3 servers running at a time for us as specified in our Auto-Scaling group when we were creating it. Hence, we have reached our maximum limit of instances running concurrently.

Name	Instance ID	Instance state	Security group name	Key name	Launch time
-	i-05f306bab83bb8ec1	Running	mysec1	debkey2	2023/05/04 16:36 GMT+5:30
-	i-04f330d210730d2a8	Running	mysec1	debkey2	2023/05/04 16:38 GMT+5:30
-	i-0a0f3aaaf93315ca2d	Running	mysec1	debkey2	2023/05/04 20:33 GMT+5:30

Hence, our Auto-Scaling Group can handle instance overloading by providing new instances to handle the overloading.

Note that our project webpage was not at all disconnected because of this.

**We have successfully created, configured and tested our Auto-Scaling Group.**

Now observe that whenever we close or terminate any instance then a new instance gets created. Hence, we cannot delete them if we want to delete them finally.

Follow these steps to cleanly remove Auto-Scaling Groups and everything related to it.

1) Go to Auto-Scaling groups and select the one which we are using.

The screenshot shows the AWS Auto Scaling Groups page. At the top, there is a header with the title 'Auto Scaling groups (1/1)' and a 'Create an Auto Scaling group' button. Below the header is a search bar labeled 'Search your Auto Scaling groups'. The main table lists one item: 'debasg1' with a status of 'debttemp1 | Version Latest'. The table includes columns for Name, Launch template/configuration, Instances, Status, Desired capacity, and Min.

2) Now click on the delete button.

3) Type delete and finally delete it.

4) Notice the status changes to Deleting. It will take some minutes to fully delete. Now go to the Load balancer section from the left side nav bar.

The screenshot shows the AWS Auto Scaling Groups and Load Balancers pages. On the left, the navigation menu is visible with the 'Auto Scaling Groups' option selected under the 'Auto Scaling' section. The top navigation bar shows the AWS logo, services, search bar, and region 'Mumbai'. The 'Auto Scaling groups' page shows the 'debasg1' group with a status of 'Deleting'. The 'Load balancers' page shows one load balancer named 'debasg1-1' with a status of 'Active'. The bottom part of the screenshot shows a modal window titled '0 load balancers selected' with the message 'Select a load balancer above.'

5) Now select the load balancer and click on the action button on the top.

The screenshot shows the AWS EC2 Load balancers page. A single load balancer, 'debasg1-1', is listed in the table. The table columns include Name, DNS name, State, VPC ID, and Availability. The 'Actions' menu on the right provides options like Edit IP address type, Edit subnets, Edit instances, Edit health check settings, Edit listener, Edit security groups, Edit load balancer attributes, Manage tags, and Delete load balancer.

Now select the Delete Load balancer option to delete it.

6) Now go to the Target Groups section.

7) Select the target group and click on the action button on the top. Select the delete option.

The screenshot shows the AWS EC2 Target groups page. A single target group, 'debasg1-1', is listed in the table. The table columns include Name, ARN, Port, Protocol, and Targets. The 'Actions' menu on the right provides options like Delete, Register targets, Edit health check settings, Edit target group attributes, Manage tags, Associate with a new load balancer, and Associate with an existing load balancer.

8) Now go to the instances page.

9) You will find that all the instances created by the Auto-Scaling group will automatically be terminated.

(If not wait for some time. Check if the Auto Scaling Group has been deleted by now.)

The screenshot shows the AWS EC2 Instances page. Three instances are listed, all in the 'Terminated' state. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS.

10) Finally check the Auto-Scaling Group section to see if it is completely removed/deleted.

The screenshot shows the AWS EC2 Auto Scaling groups page. It displays a message stating 'No Auto Scaling groups found in the current region'. A 'Create an Auto Scaling group' button is available. Below this, it says '0 Auto Scaling groups selected' and 'Select an Auto Scaling group'.

Everything was deleted successfully.

**Hence, we successfully deleted our Auto-Scaling Group and all the load balancers, target groups, instances that was created.**

# ASSIGNMENT-12

**Problem Statement:** Deploy a project from GitHub to EC2 without using Port.

## Procedures:

1. Sign-in to AWS console.
2. Go to the EC2 dashboard. Now go to the instances page.
3. Click on the create new instance button.
4. Now create an EC2 server using the Security Group created earlier and enter the user data  
**(Refer to Ass10)**

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  Add additional tags

**Recent** **Quick Start**

Amazon Linux  macOS  Ubuntu  Windows  Red Hat  >  Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type Free tier eligible

ami-02eb7a4783e7e9317 (64-bit (x86)) / ami-0a5dcfffb7af3fc9 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Description  
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture  AMI ID: ami-02eb7a4783e7e9317 Verified provider

**Instance type** [Info](#)

Instance type  Free tier eligible

t2.micro 1 vCPU 1 GiB Memory Current generation: true  
Family: t2 On-Demand Linux pricing: 0.0124 USD per Hour  
On-Demand Windows pricing: 0.017 USD per Hour  
On-Demand RHEL pricing: 0.0724 USD per Hour  
On-Demand SUSE pricing: 0.0124 USD per Hour

All generations  Compare instance types

**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

**Network settings** [Info](#)

Network [Info](#)  
vpc-0a33deec3fd6dc096

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

Security groups [Info](#)  
 Compare security group rules

mysec1 sg-0493398d43b761e55   
VPC: vpc-0a33deec3fd6dc096

**Advanced details** [Info](#)

User data - optional [Info](#)  
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/DebrupPramanik/myRepoV1.git
cd myRepoV1
npm install
node index.js
```

[Launch instance](#)

User data has already been base64 encoded

[Review commands](#)

## 5. Create the instance and click on the instance after creation.

Instances (1) <a href="#">Info</a>		<a href="#">C</a>	Connect	Instance state	Actions	<a href="#">Launch instances</a>	▼	
<a href="#">Find instance by attribute or tag (case-sensitive)</a>								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS

debser1 i-0204a46d8ffd384cb Running Initialization t2.micro No alarms + ap-south-1 ec2-65-2-169-22

## 6. Copy the public IPv4 address and paste it in another browser. The nginx homepage will show up.

**Instance summary for i-0204a46d8ffd384cb (debser1) [Info](#)**  
Updated less than a minute ago

Instance ID <a href="#">i-0204a46d8ffd384cb (debser1)</a>	Public IPv4 address <a href="#">open address</a>	Private IPv4 addresses <a href="#">172.31.38.17</a>
IPv6 address -	Instance state <span style="color: green;">Running</span>	Public IPv4 DNS <a href="#">ec2-65-2-169-220.ap-south-1.compute.amazonaws.com</a>   <a href="#">open address</a>
Hostname type IP name: ip-172-31-38-17.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) <a href="#">ip-172-31-38-17.ap-south-1.compute.internal</a>	



### Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](#). Commercial support is available at [nginx.com](#).

*Thank you for using nginx.*

Our server is working perfectly. Note, in previous assignments we used to connect to our project webpages using port no. However, in this exercise we are going to access our project webpage without using any port no.

## 7. Copy the Public IPv4 address of the server instance and use this to connect it to the server using Bitvise SSH client. (Refer Ass7)

**Default profile**

- [Save profile as](#)
- [Bitvise SSH Server Control Panel](#)
- [New terminal console](#)

Login	Options	Terminal	RDP	SFTP	Services	C2S	S2C	SSH	Notes	About
<b>Server</b> Host: <input type="text"/> Port: <input type="text"/> <input type="checkbox"/> Enable obfuscation Obfuscation keyword: <input type="text"/>	<b>Authentication</b> Username: <input type="text"/> ubuntu Initial method: <input type="text"/> publickey Client key: <input type="text"/> Global 2 Passphrase: <input type="text"/> Elevation: <input type="text"/> Default									
<b>Kerberos</b> SPN: <input type="text"/> <input type="checkbox"/> GSS/Kerberos key exchange <input type="checkbox"/> Request delegation <input checked="" type="checkbox"/> gssapi-keyex authentication										

**8. Now open the terminal in Bitvise.**

```
ubuntu@65.2.169.220:22 - Bitvise xterm - ubuntu@ip-172-31-38-17: ~
Last login: Wed May 10 12:14:04 2023 from 150.129.133.232
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-38-17:~$
```

**9. Enter the following commands in it.**

→ **pwd**

```
ubuntu@ip-172-31-38-17:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-38-17:~$
```

(To check current directory)

→ **cd /**

```
ubuntu@ip-172-31-38-17:~$ cd /
```

(To go to root folder)

→ **pwd**

```
ubuntu@ip-172-31-38-17:/~$ pwd
/
```

→ **cd /etc/nginx/sites-available/**

```
ubuntu@ip-172-31-38-17:/~$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-38-17:/etc/nginx/sites-available$
```

(To open the sites-available directory under nginx)

→ **sudo nano default**

```
ubuntu@ip-172-31-38-17:/etc/nginx/sites-available$ sudo nano default
GNU nano 6.2                               default
##
## You should look at the following URL's in order to grasp a solid understanding
## of Nginx configuration files in order to fully unleash the power of Nginx.
## https://www.nginx.com/resources/wiki/start/
## https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
## https://wiki.debian.org/Nginx/DirectoryStructure
##
## In most cases, administrators will remove this file from sites-enabled/ and
## leave it as reference inside of sites-available where it will continue to be
## updated by the nginx packaging team.
##
## This file will automatically load configuration files provided by other
## applications, such as Drupal or Wordpress. These applications will be made
## available underneath a path with that package name, such as /drupal18.
##
## Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##
```

```
# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
```

(To open the default file in the nano editor)

**10. After opening the default file in the nano editor, search for the location / {}. It should be after server\_name\_;**

```
server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}
```

11. Comment out the location block and each and every line inside the block.

```
server_name _;

#location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    #try_files $uri $uri/ =404;
}
```

12. Now paste the following code just under the closing curly bracket.

```
location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'Upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
```

```
server_name _;

#location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    #try_files $uri $uri/ =404;
}
location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'Upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
```

13. Now save it by Ctrl+X and exit nano editor.

14. You will be reverted back to the terminal. Type the following command....

→ sudo systemctl restart nginx

```
ubuntu@ip-172-31-2-192:/etc/nginx/sites-available$ sudo systemctl start nginx
```

15. Now paste the public IPv4 address in your browser. Now press Enter. Our project page will show up without entering our port no.

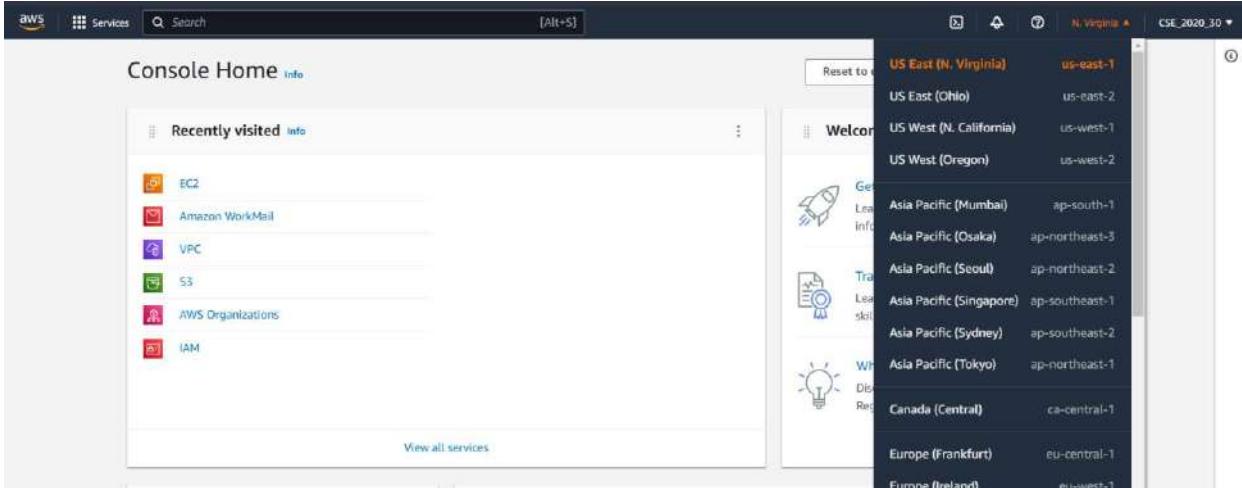


We have successfully deployed a project from GitHub to EC2 without using port.

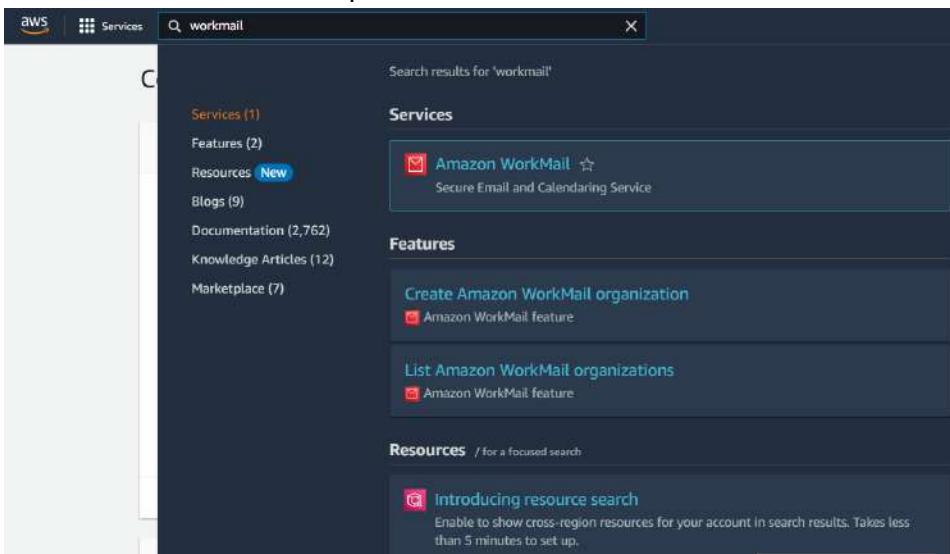
# ASSIGNMENT-13

**Problem Statement:** Create a workmail for your profile.

1. Sign-in to your AWS console.
2. Change your region to N.Virginia from the dropdown on the left-side of the username in the top bar of the console homepage.



3. Search for workmail.
4. Select the AWS workmail option.



5. Click on the create organization button.



## 6. Select the free test domain option under email domain.

Email domain [Info](#)  
Select the domain to use for email addresses in your organization.

Existing Route 53 domain  
Select a domain name that you manage with a Route 53 hosted zone.

New Route 53 domain  
Register a new Route 53 domain name to use with Amazon WorkMail.

External domain  
Enter a domain name that you manage with an external DNS provider.

Free test domain  
Use a free testing domain provided by Amazon WorkMail. You can add a domain later.

7. Give the organization name/ alias. Make sure it is unique in the region.

8. Next click on create organization button.

9. Wait for two minutes.

10. After waiting the organization will become active. Now click on the organization name.

Amazon WorkMail > Organizations

Organizations (1) [Info](#)

Find organizations

papinoob	<input type="radio"/>
Organization ID	m-86566ac1147549f187484e2793ace4ad
Default domain	papinoob.awsapps.com
State	Active

You are using the test domain as your default domain. We recommend that you add a custom domain and set it as the default domain.

Manage domains

Amazon WorkMail > Organizations > papinoob

**papinoob**

**Organization details** [Info](#)

Organization ID	<a href="#">m-86566ac1147549f187484e2793ace4ad</a>	State	Active	Directory type	WorkMail directory
ARN	<a href="#">arn:aws:workmail:us-east-1:728364961341:organization/m-86566ac1147549f187484e2793ace4ad</a>	Date created	May 10, 2023 at 20:51 (UTC+5:30)	Directory ID	<a href="#">d-906794a9b0</a>
		Default domain	<a href="#">Info</a>		<a href="#">papinoob.awsapps.com</a>

**User login**

Desktop or mobile apps	<a href="#">WorkMail documentation for setting up email clients</a>	Amazon WorkMail web application	<a href="https://papinoob.awsapps.com/mail">https://papinoob.awsapps.com/mail</a>
------------------------	---	---------------------------------	---

11. Now go to the Users section under Organization tab on the left side nav bar.

Amazon WorkMail

Organizations

What's new

**Organization**

Users

Groups

Resources

**12. Click on the create user button.**

Amazon WorkMail > Organizations > papinoob > Users

Users (0) Info

Search users

Display name | User name | Primary email address | State

No users to display.

Create user

**13. Give username. Then display name.**

Amazon WorkMail > Organizations > papinoob > Users > Create user

Create a user Info

Add a user to your Amazon WorkMail organization.

User details

User name  
The user name enables the user to login to the Amazon WorkMail webmail.  
david

User name can only contain the following characters: a-z, A-Z, 0-9, \_, (underscore), - (hyphen) and @.

First name - optional

Last name - optional

Display name  
The name by which the user is presented in the system.  
david

**14. Then provide the password.**

Email setup

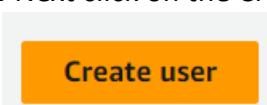
Email address  
Primary email address to be used for this user.  
david @ papinoob.awsapps.com

Password  
Password for the user to log in with.  
\*\*\*\*\*

Passwords must be an 8-character minimum with at least one character from three of these four categories: lowercase, uppercase, numeric, and special characters.

Repeat password  
\*\*\*\*\*

**15. Next click on the Create User button.**



**16. After the user is created return back to your organization page using the following link.**

aws Services Search [Alt+S] N. Virginia CSE\_2020

Amazon WorkMail

Successfully created david.

Amazon WorkMail > Organizations > papinoob > Users

Users (1) Info

Search users

Display name | User name | Primary email address | State

david | david | david@david.papinoob.awsapps.com | Enabled

Create user

**17. Click on the organization name.**

papinoob

Organization ID  
m-86566ac1147549f187484e2793ace4ad

Default domain  
papinoob.awsapps.com

State  
Active

**18.** Now click on the Amazon Workmail web application link.

The screenshot shows the 'Organization details' section of the AWS WorkMail console. It includes fields for Organization ID (m-86566ac1147549f187484e2793ace4ad), State (Active), Directory type (WorkMail directory), ARN (arn:aws:workmail:us-east-1:728364961341:organization/m-86566ac1147549f187484e2793ace4ad), Date created (May 10, 2023 at 20:51 (UTC+5:30)), Directory ID (d-906794a9b0), Default domain (papinoob.awsapps.com), and User login links for desktop/mobile apps and the Amazon WorkMail web application.

**19.** Enter the credentials of the user you just created in it.



Please log in with your papinoob credentials

Username (not email address)

Username

Remember username

Password

Password

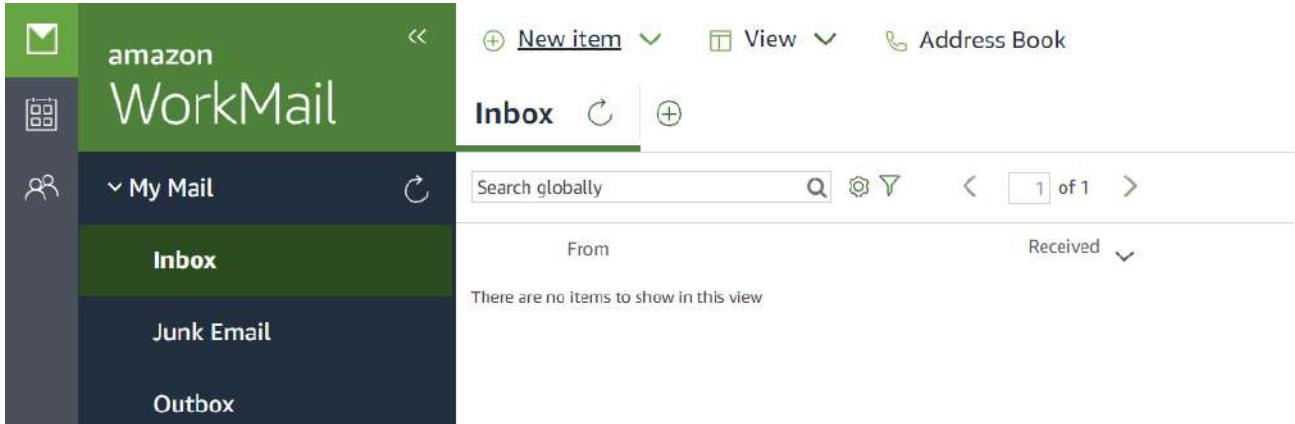
**Sign In**

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

**20.** Our mail will be opened as the entered user.

The screenshot shows the AWS WorkMail inbox interface. The left sidebar lists 'My Mail' sections: Inbox (selected), Junk Email, Outbox, Drafts, Sent Items, Deleted Items, and RSS Feeds. The main area displays the 'Inbox' view with a search bar and a message stating 'There are no items to show in this view'. The top right corner shows the user's name (david) and various navigation icons.

- 21.** Now let's test our mail server by seeing whether it can send and receive mail. Send some mail to your Gmail from this by creating a mail using the new item option.



- 22.** Select new email. Now write an email to your Gmail address.

david Sent on: Today, 9:11 pm

Hello World

Hello,  
This is David from AWS Workmail.

- 23.** Check your Gmail.

Hello World Inbox

 david <david@papinoob.awsapps.com>  
to me ▾

Hello,  
This is David from AWS Workmail.

← Reply → Forward

We received it.

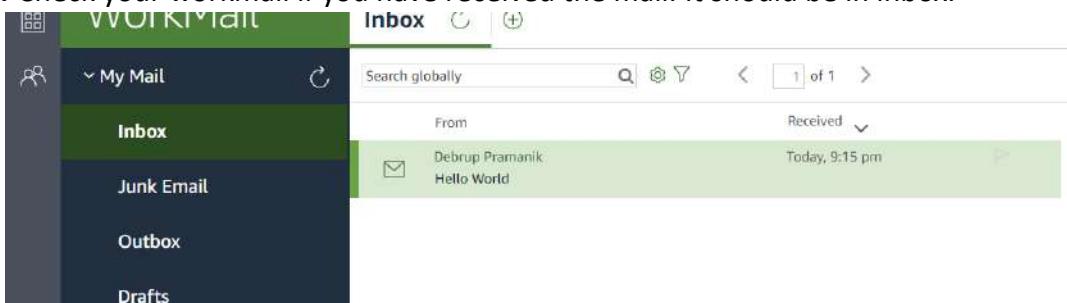
- 24.** Now send something back to the workmail.

 Debrup Pramanik <debrup202002@gmail.com>  
to david ▾

Hello,  
This is Debrup from Gmail. I received your request.

← Reply → Forward

- 25.** Check your workmail if you have received the mail. It should be in inbox.



Hello World

To david

---

Hello,  
This is Debrup from Gmail. I received your request.

We have successfully received a mail.

**Hence our mail is working perfectly fine.**

**We have successfully created a workmail for our profile.**

# ASSIGNMENT – 14

## Problem Statement: Create an elastic IP for an instance.

1. Sign-in to your AWS console.
2. Create an EC2 instance. (We do not need any user-data or any custom security group for this assignment)

The screenshot shows the AWS EC2 Instances list. There is one instance named "debserver1" which is "Running". The Public IPv4 address is listed as 3.110.83.71. The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

3. After the instance gets created click on it. Copy the public IPv4 address and paste it in a simple text file anywhere in your pc.

The screenshot shows the Instance summary for the instance "debserver1". It displays the Public IPv4 address as 3.110.83.71. A large black box highlights this address. The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

4. Now go back to the instances list and select our instance.

The screenshot shows the AWS EC2 Instances list with the instance "debserver1" selected. The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

5. After selection click on the Instance state button and click on the Stop Instance option.

The screenshot shows the AWS EC2 Instances list with the instance "debserver1" selected. The "Instance state" dropdown menu is open, showing options: "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

6. Wait for few seconds.

7. Now again select the instance and click on the Instance state button. Now click on the start instance button.

The screenshot shows the AWS EC2 Instances list with the instance "debserver1" selected. The "Instance state" dropdown menu is open, showing options: "Force stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

8. Click on the instance and copy the IPv4 address again and paste it in the same text file.

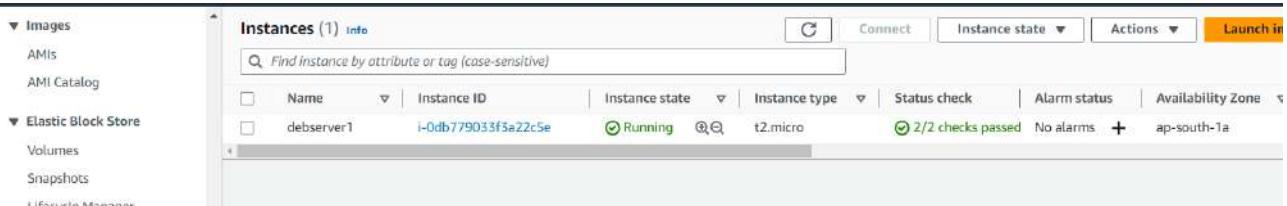
The screenshot shows the Instance summary for the instance "debserver1". It displays the Public IPv4 address as 43.205.95.224. A large black box highlights this address. The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

## 9. Now compare both the new and old IP address and notice that they are not the same.

```
3.110.83.71  
43.205.95.224
```

So even if we stop and restart our same instance it changes its public IPv4 address. This may not be desirable in some situations. So, to ensure that our instance does not change its public IPv4 address under any circumstances, we need to create an Elastic IP and associate/bind the instance to it. After that it will always be assigned the same Elastic IP as its public IPv4 address (static) all the time.

## 10. For creating an Elastic IP, we need to go scroll down the left side Nav bar and find the Network and security section.



## 11. Under it click on the Elastic IPs option.

### ▼ Network & Security

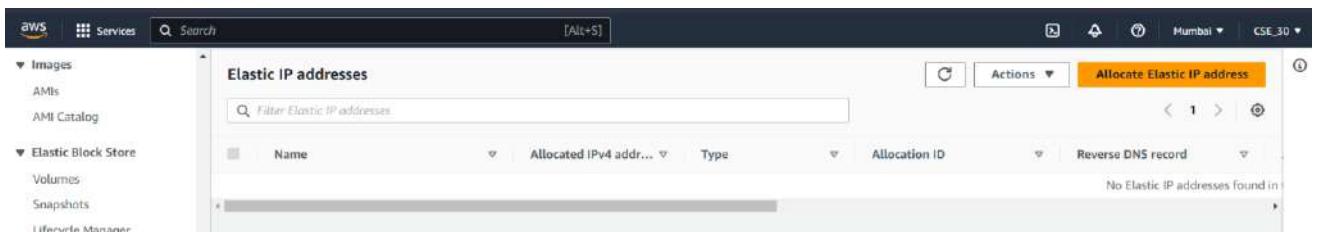
Security Groups

Elastic IPs

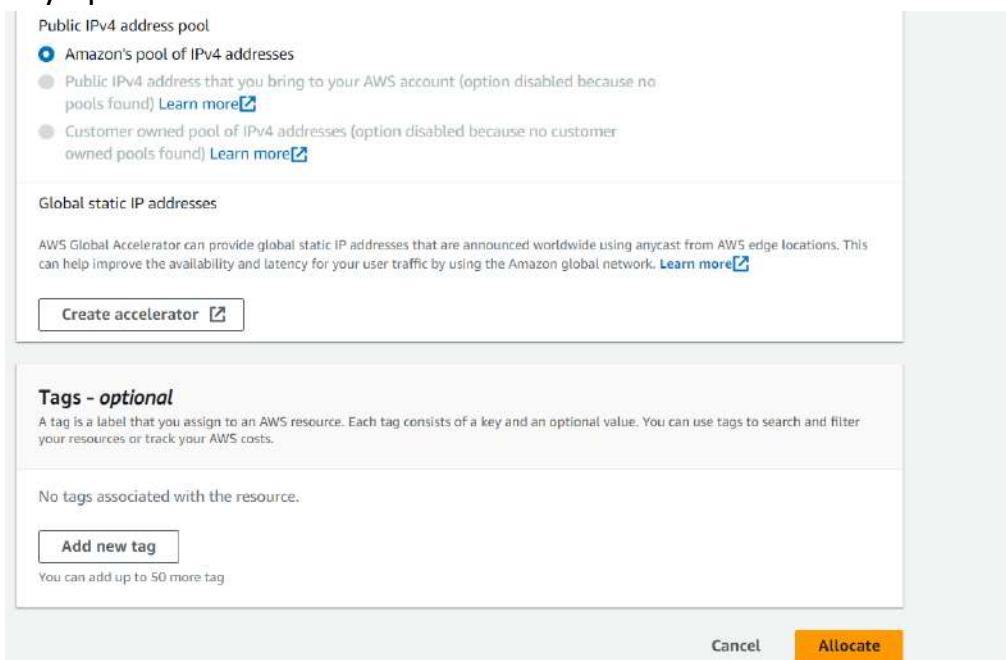
Placement Groups

Key Pairs

Network Interfaces



## 12. Now, click on the Allocate Elastic IP address button on the right side. No need to change any options. Just click on the Allocate button.



**13. Now click on the Elastic IP address (in blue).**

Elastic IP addresses (1/1)					
Name		Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record
<input checked="" type="checkbox"/>	-	<a href="#">3.7.231.115</a>	Public IP	eipalloc-0864617bd007a2f9b	-

**14. Next click on the Associate Elastic IP address button.**

EC2 > Elastic IP addresses > 3.7.231.115

**3.7.231.115**

**Actions** **Associate Elastic IP address**

**Summary**

Allocated IPv4 address <a href="#">3.7.231.115</a>	Type <a href="#">Public IP</a>	Allocation ID <a href="#">eipalloc-0864617bd007a2f9b</a>	Reverse DNS record -
Association ID -	Scope <a href="#">VPC</a>	Associated instance ID -	Private IP address -
Network interface ID -	Network interface owner account ID -	Public DNS -	NAT Gateway ID -
Address pool <a href="#">Amazon</a>	Network Border Group <a href="#">ap-south-1</a>		

**Tags (0)** **Manage tags**

Key	Value

**15. Choose your instance you want to associate with it.**

**16. Keep the Private IP address as specified in the dropdown when clicking for the Private Address.**

**17. Select the Allow Elastic IP to be reassociated option if we want to reuse it again for another instance.**

**Elastic IP address: 3.7.231.115**

Resource type  
Choose the type of resource with which to associate the Elastic IP address.

Instance  
 Network interface

**⚠ If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. [Learn more](#)**

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

**Instance**

**Private IP address**  
The private IP address with which to associate the Elastic IP address.

**Reassociation**  
Specify whether the Elastic IP address can be reassigned to a different resource if it is already associated with a resource.

Allow this Elastic IP address to be reassigned

18. Now click the associate button.
19. The Elastic IP should have been successfully associated with the instance.
20. To check it go back to the instances page. Click on the Instance and see the Public IPv4 address and the Elastic IP address. They should be same. Also notice that the public IPv4 address has turned into a hyperlink to the Elastic IP page.

Instance summary for i-0db779033f3a22c5e (debserv1) <a href="#">Info</a>		Actions
Updated less than a minute ago		
Instance ID <a href="#">i-0db779033f3a22c5e (debserv1)</a>	Public IPv4 address <a href="#">3.7.231.115   open address</a>	Private IPv4 addresses <a href="#">172.31.40.195</a>
IPv6 address -	Instance state <span style="color: green;">Running</span>	Public IPv4 DNS <a href="#">ec2-3-7-231-115.ap-south-1.compute.amazonaws.com   open address</a>
Hostname type IP name: ip-172-31-40-195.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) <a href="#">ip-172-31-40-195.ap-south-1.compute.internal</a>	Elastic IP addresses <a href="#">3.7.231.115 [Public IP]</a>
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding <a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>   Learn more
Auto-assigned IP address -	VPC ID <a href="#">vpc-0a33deec3fd6dc096</a>	Auto Scaling Group name -
IAM Role -	Subnet ID <a href="#">subnet-0bbe74a9835a07e38</a>	

**Now stop and restart the instance and see if the public IPv4 address changes or not. It will not change.**

**Hence, we have successfully created an Elastic IP for an instance.**

To delete the Elastic IP, follow these steps:

1. Click on the Elastic IP.
2. Click on the actions button.

EC2 > Elastic IP addresses > 3.7.231.115

3.7.231.115			Actions									
<b>Summary</b> <table border="1"> <tr> <td>Allocated IPv4 address <a href="#">3.7.231.115</a></td> <td>Type <a href="#">Public IP</a></td> <td>Allocation ID <a href="#">eipalloc-0864617bd007a2f9b</a></td> </tr> <tr> <td>Association ID <a href="#">eipassoc-053b5539deac45cb4</a></td> <td>Scope <a href="#">VPC</a></td> <td>Associated instance ID <a href="#">i-0db779033f3a22c5e</a></td> </tr> <tr> <td>Network interface ID <a href="#">eni-0d87560629582fe63</a></td> <td>Network interface owner account ID <a href="#">728364961341</a></td> <td>Public DNS <a href="#">ec2-3-7-231-115.ap-south-1.compute.amazonaws.com</a></td> </tr> </table>			Allocated IPv4 address <a href="#">3.7.231.115</a>	Type <a href="#">Public IP</a>	Allocation ID <a href="#">eipalloc-0864617bd007a2f9b</a>	Association ID <a href="#">eipassoc-053b5539deac45cb4</a>	Scope <a href="#">VPC</a>	Associated instance ID <a href="#">i-0db779033f3a22c5e</a>	Network interface ID <a href="#">eni-0d87560629582fe63</a>	Network interface owner account ID <a href="#">728364961341</a>	Public DNS <a href="#">ec2-3-7-231-115.ap-south-1.compute.amazonaws.com</a>	<a href="#">Associate Elastic IP address</a> <ul style="list-style-type: none"> <li><a href="#">Release Elastic IP addresses</a></li> <li><a href="#">Disassociate Elastic IP address</a></li> <li><a href="#">Update reverse DNS</a></li> <li><a href="#">Enable transfers</a></li> <li><a href="#">Disable transfers</a></li> <li><a href="#">Accept transfers</a></li> </ul>
Allocated IPv4 address <a href="#">3.7.231.115</a>	Type <a href="#">Public IP</a>	Allocation ID <a href="#">eipalloc-0864617bd007a2f9b</a>										
Association ID <a href="#">eipassoc-053b5539deac45cb4</a>	Scope <a href="#">VPC</a>	Associated instance ID <a href="#">i-0db779033f3a22c5e</a>										
Network interface ID <a href="#">eni-0d87560629582fe63</a>	Network interface owner account ID <a href="#">728364961341</a>	Public DNS <a href="#">ec2-3-7-231-115.ap-south-1.compute.amazonaws.com</a>										

3. From the drop-down menu select Disassociate Elastic IP address. Then again click on disassociate on the pop-up.
4. Next again click on the Actions button and this time select Release Elastic IP address.

EC2 > Elastic IP addresses > 3.7.231.115

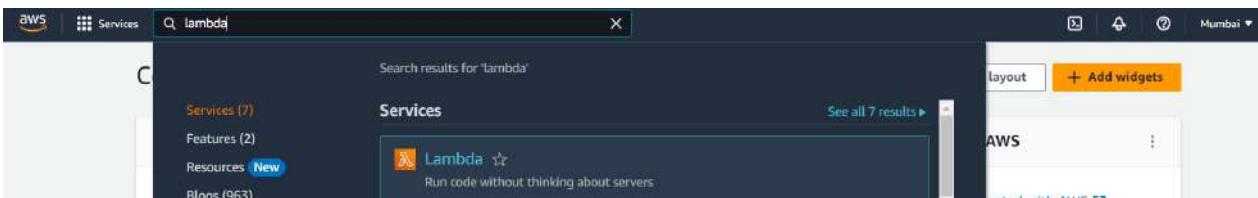
3.7.231.115		Actions
		<a href="#">Associate Elastic IP address</a> <ul style="list-style-type: none"> <li><a href="#">Release Elastic IP addresses</a></li> </ul>

5. Now you can go back to your instance and see that the IPv4 address has already changed to a random one and it has no Elastic IP address associated with it. Now you can terminate the instance.

# ASSIGNMENT – 15

**Problem Statement:** Create a serverless computing service.

1. Sign-in to your AWS console.
2. Search for Lambda



Click on the first result named Lambda.

3. Now click on the Create Function button on the top right corner.



4. Select Author from scratch option.



5. Give the name of the function.



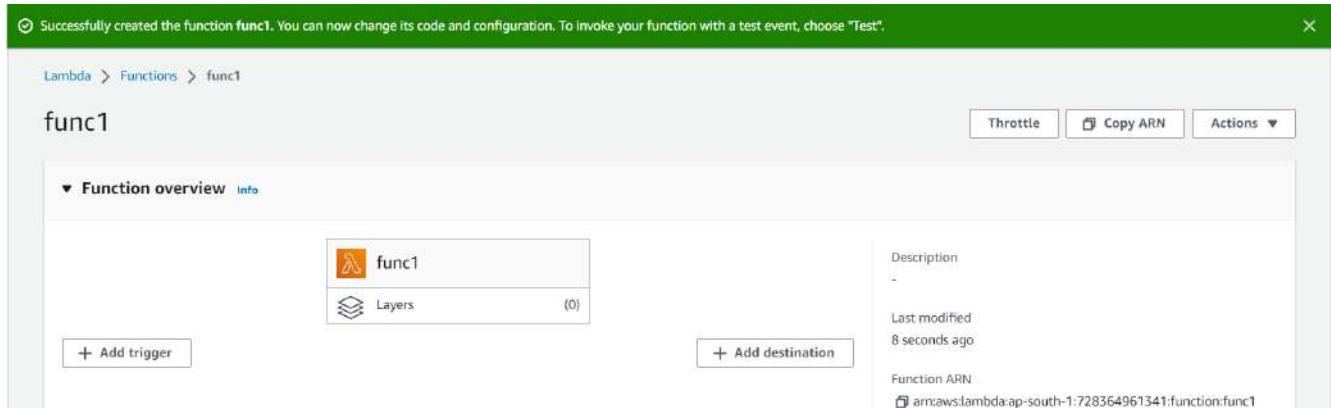
6. Choose Node.js as the Runtime. (No need to change architecture.)



7. Now click on the create function button



8. Now scroll-down to the code section of your newly created function.



9. Change the string in the code to be displayed.

The screenshot shows the AWS Lambda code editor interface. At the top, there are tabs for Code, Test, Monitor, Configuration, Aliases, and Versions. Below the tabs, the title "Code source" and "Info" are displayed. A toolbar with File, Edit, Find, View, Go, Tools, Window, Test, Deploy, and settings icons is visible. On the left, an "Environment" sidebar shows a "func1" folder containing "index.mjs". The main editor window shows the following code:

```
1 export const handler = async(event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from Lambda!'),
6     };
7     return response;
8 };
9 
```

We changed it to this.....

The screenshot shows the AWS Lambda code editor interface with the "File" option highlighted in the menu bar. The main editor window shows the modified code:

```
1 export const handler = async(event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from AWS !!!!!'),
6     };
7     return response;
8 };
9 
```

10. Now go to File option and click on save to save the changes.

The screenshot shows the AWS Lambda code editor interface with the "File" option highlighted in the menu bar. The "Save" option is selected, indicated by a blue background. The main editor window shows the modified code:

```
1 export const handler = async(event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from AWS !!!!!'),
6     };
7     return response;
8 };
9 
```

11. Now click on the Test button.

The screenshot shows the AWS Lambda code editor interface with the "Test" button highlighted in the toolbar. The main editor window shows the modified code:

```
1 export const handler = async(event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from AWS !!!!!'),
6     };
7     return response;
8 };
9 
```

12. Select Create New Event. Then give a name. Then click on save.

The screenshot shows the "Configure test event" dialog box. It includes a description of what a test event is, instructions to invoke the function without saving an event, and sections for selecting the test event action (Create new event or Edit saved event), entering an event name (e1), and setting sharing options (Private).

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

Create new event     Edit saved event

Event name

e1

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

**Event JSON**

[Format JSON](#)

1  
2  
3

Cancel

[Save](#)

13. After saving. Now click on the Deploy button.

**Code source** [Info](#)

File Edit Find View Go Tools Window

[Test](#)

[Deploy](#)

14. After successful deployment a message will pop as mentioned below and the deploy button will be locked out, indicating that our Function has been successfully deployed.

 Successfully updated the function func1. X

15. Now go to the configuration tab.

Code Test Monitor **Configuration** Aliases Versions

**General configuration**

Triggers

Permissions

Destinations

Function URL

Environment variables

**General configuration** [Info](#)

Description

-

Memory

128 MB

Timeout

0 min 3 sec

SnapStart [Info](#)

None

16. Click on the Function URL option in the left side Bar.

**General configuration**

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

Monitoring and operations tools

17. Click on Create function URL.

[Function URL](#) [Info](#)

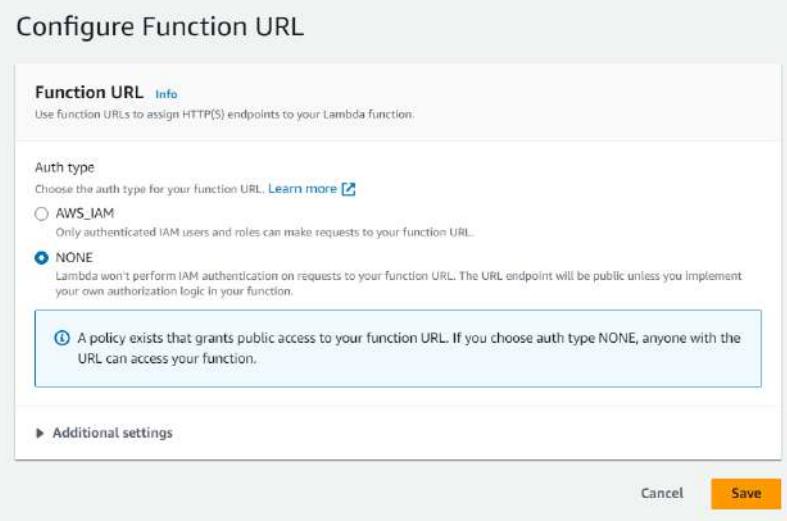
[Create function URL](#)

No Function URL

No Function URL is configured.

[Create function URL](#)

**18. Choose NONE and click on the save button.**



**19. Now copy the newly created Function URL and paste it in a different browser.**

Your changes have been saved.

Function overview

func1

Description

Last modified

6 minutes ago

Function ARN

arn:aws:lambda:ap-south-1:728364961341:function:func1

Function URL

https://ejkt6cr7wkr5hqrb2iavadyhe0thqki.lambda-url.ap-south-1.on.aws/

https://ejkt6cr7wkr5hqrb2iavadyhe0thqki.lambda-url.ap-south-1.on.aws

"Hello from AWS !!!!"

We have successfully Created a Serverless Computing service.

**To delete the Lambda Function, follow these steps:**

1. Click on the Actions button on the top right side.

Your changes have been saved.

Lambda > Functions > func1

func1

Throttle

Copy ARN

Actions

Function overview

func1

Description

Last modified

13 minutes ago

Function ARN

arn:aws:lambda:ap-south-1:728364961341:function:func1

Function URL

https://ejkt6cr7wkr5hqrb2iavadyhe0thqki.lambda-url.ap-south-1.on.aws/

2. Select the Delete function option and then click on delete button in the pop-up.

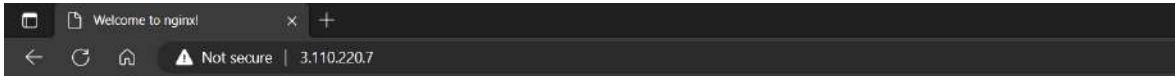
# ASSIGNMENT – 16

## Problem Statement: Manage Amazon DNS service and run a project using domain-name and URL

1. Sign-in to your AWS console.
2. Create an instance with custom security group and user data (Refer Ass10)
3. Click on the instance

The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for 'Info', 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below is a search bar and a 'Clear filters' button. A table lists one instance: 'Name' (debserverX), 'Instance ID' (i-0da0c914af8583268), 'Instance state' (Running), 'Status check' (2/2 checks passed), 'Alarm status' (No alarms), 'Availability Zone' (ap-south-1a), and 'Public IPv4 DN' (ec2-3-110-220-). There are also 'Edit' and 'Delete' icons for the instance.

4. Copy the public IPv4 address and paste it in another browser.



### Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

5. Check if the project webpage is accessible by appending :4000 to your address.



Our EC2 instance works as intended. However, to access our webpage one always requires the public IPv4 address of our server instance which is very complicated/less accessible for end-users of our webpage/web application.

So, to make it easier for our end-users, we need to bind a domain name to the server instance. Now anyone can use the domain name and the URL to access our project.

6. Search Route 53 in the search bar of AWS console. Select the first result.

A screenshot of the AWS Route 53 dashboard. The search bar at the top has 'route 53' typed into it. On the left, there's a sidebar with 'EC2 Dashboard', 'Features (19)', 'Resources (New)', 'Blogs (249)', 'Documentation (16,536)', 'Knowledge Articles (16)', 'Tutorials (2)', 'Events (1)', and 'Marketplace (709)'. The main area shows 'Services' with 'Route 53' selected. To the right, there's a 'Account attributes' panel showing 'Supported platforms' (VPC), 'Default VPC' (vpc-0a33dec3fd6dc096), 'Settings' (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments), and an 'Explore AWS' section.

We require a registered Domain name for this assignment. So, after obtaining one (free or paid) go to the Webpage of your Domain provider and log-in to your account where you can find all the details of your purchased Domains.

This may vary from site to site, so you will have to do this based on what site you are using.  
We (for now) will be using GoDaddy.com, because we have purchased a Domain from them.

7. After Reaching the Route 53 dashboard click on the Create Hosted Zone button.

## Route 53 Dashboard [Info](#)

**DNS management**  
A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.

**Traffic management**  
A visual tool that lets you easily create policies for multiple endpoints in complex configurations.

**Availability monitoring**  
Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.

**Domain registration**  
A domain is the name, such as example.com, that your users use to access your application.

**Create hosted zone** **Create policy** **Create health check** **Register domain**

**Register domain**

Find and register an available domain, or transfer your existing domains to Route 53.

Enter a domain name

Each label (each part between dots) can be up to 63 characters long and must start with a-z or 0-9. Maximum length: 255 characters, including dots. Valid characters: a-z, 0-9, and - (hyphen).

**Check**

Alternatively, you can go to hosted zones from the left-side bar and then select create hosted zone option.

- Now, copy your Domain name from your Domain providers website. Here we used GoDaddy.com. Paste the domain name in the given field in Hosted Zone configuration page.

1 domain

**Domain Name** ↑

[debrup.co.in](#) ...

**Create hosted zone [Info](#)**

**Hosted zone configuration**  
A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

**Domain name [Info](#)**  
This is the name of the domain that you want to route traffic for.

Valid characters: a-z, 0-9, ! " # \$ % & ' ( ) \* + , - / ; : < = > ? @ [ \ ] ^ \_ [ ] , ~

**Description - optional [Info](#)**  
This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 0/256

**Type [Info](#)**  
The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

**Public hosted zone**  
A public hosted zone determines how traffic is routed on the internet.

**Private hosted zone**  
A private hosted zone determines how traffic is routed within an Amazon VPC.

- Now scroll-down and click on the Create Hosted Zone button.

- Now click on the Create a record button.

Route 53 > Hosted zones > debrup.co.in

**Public debrup.co.in [Info](#)**

**Hosted zone details**

**Records (2)** **DNSSEC signing** **Hosted zone tags (0)**

**Records (2) [Info](#)**  
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Record ...	Type	Routing policy	Alias	Value/Route traffic to	TTL (\$...)	Health ...
debrup.co.in	NS	Simple	-	No ns-758.awsdns-30.net. ns-1483.awsdns-57.org. ns-2015.awsdns-59.co.uk. ns-327.awsdns-40.com.	172800	-
debrup.co.in	SOA	Simple	-	No ns-758.awsdns-30.net. awsd...	900	-

**Create record**

- Follow these Steps:

- Do not give any name. Keep the record name blank.
- Keep record type as it is. No change required.
- Under the value, copy and paste your server instance public IPv4 address which you want to route to using your DNS.
- Then click on create records button

Quick create record

**Record 1**

Record name: **Info** subdomain debrup.co.in  
Keep blank to create a record for the root domain.

Record type: **Info** A – Routes traffic to an IPv4 address and some AWS resources

Value: **Info** 192.0.2.235  
Enter multiple values on separate lines.

TTL (seconds): **Info** 500 1m 1h 1d Routing policy: **Info** Simple routing  
Recommended values: 60 to 172800 (two days)

Add another record Cancel Create records

## 12. Now again click on the Create Record button like the previous step.

- But this time give the record name as → **www**
- Select Record type as **CNAME**
- In the text box under value, write the full domain-name there. (For example: example.com)
- Click on create records button

Quick create record

**Record 1**

Record name: **Info** www debrup.co.in  
Keep blank to create a record for the root domain.

Record type: **Info** CNAME – Routes traffic to another domain name and to some AWS reso...  
CNAME

Value: **Info** debrup.co.in  
Enter multiple values on separate lines.

TTL (seconds): **Info** 500 1m 1h 1d Routing policy: **Info** Simple routing  
Recommended values: 60 to 172800 (two days)

Add another record Cancel Create records

## 13. Now select the record with type nameserver (NS).

Hosted zone details

Records (4) Info

Records (1/4) Info

Record details

Record name: debrup.co.in

Record type: NS

Value: ns-758.awsdns-30.net., ns-1483.awsdns-57.org., ns-2015.awsdns-59.co.uk., ns-327.awsdns-40.com.

Alias: No

TTL (seconds): 172800

The values seen on the right-hand side are required for the next steps.

## 14. Now go to your Domain providers webpage. Go to your purchased Domains settings.

< Domain Portfolio

debrup.co.in

Overview

DNS

Products

15. Click on DNS section. (This may vary from provider to provider)

16. Click on the nameservers option.

< Domain Portfolio

debrup.co.in

Overview

**DNS**

Products

DNS Records

Forwarding

**Nameservers**

Premium DNS

Hostnames

17. Click on the Change nameservers and add here all the values opened in the Route 53 page.

- a. Select use my own nameservers option.
- b. Add nameservers.
- c. Then click on the save button.

Nameservers determine where your DNS is hosted and where you add, edit or delete your DNS records.

Using default nameservers

Change Nameservers

Nameservers ⓘ

X

Edit nameservers

Choose nameservers for debrup.co.in

GoDaddy Nameservers (recommended)

I'll use my own nameservers

ns-758.awsdns-30.net

ns-1483.awsdns-57.org

ns-2015.awsdns-59.co.uk

ns-327.awsdns-40.com

Add Nameserver

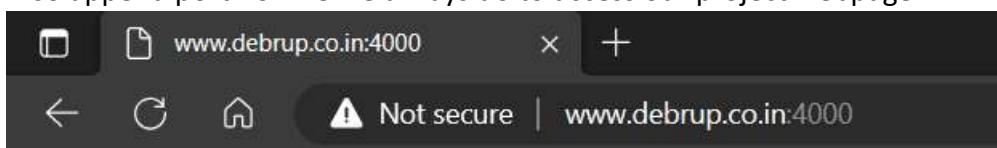
Save Cancel

18. Wait for few minutes.

19. Now try searching from any browser using your domain name with www.

(For example: [www.example.com](http://www.example.com))

20. Also append port no. like we always do to access our project webpage.



Hello. My Name is Spider-Man!!! Nice to meet You!!!

We have successfully run our project using our custom domain-name and URL.