Name: Shivshankar J Ghyar
PRN: 202201040031
Batch: CCF1

CLOUD ACTIVITY 01

ACTIVITY DETAILS	
Name of Activity	Creating networking infrastructure for launching web application on AWS
Activity Details	Create a complete networking infrastructure for launching a web application using AWS services using command line tools

1.AWS Configure through CLI

```
C:\Users\Shivshankar>aws configure
AWS Access Key ID [****************DK5E]: AKIA2NK3YLXZJRSNDK5E
AWS Secret Access Key [****************C7H3]: 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
Default region name [ap-south-1]:
Default output format [json]:
```

2.VPC Creation

```
C:\Users\Shivshankar>aws ec2 create-vpc --cidr-block 10.0.0.0/16
{
    "Vpc": {
        "OwnerId": "715841363442",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-08cd482ae7d2b72
a3",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": false,
        "VpcId": "vpc-0f0fb90e22f4d7bf4",
        "State": "pending",
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-0b3f715aec34ad700"
    }
}
```

3.Creating Subnets:

```
C:\Users\Shivshankar>aws ec2 create-subnet --vpc-id vpc-0f0fb90e22f4d7bf4 --cidr-block 10.0.1.0/24 --av
ailability-zone ap-south-1a
{
    "Subnet": {
        "AvailabilityZoneId": "aps1-az1",
        "OwnerId": "715841363442",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:ap-south-1:715841363442:subnet/subnet-0d2570c9fd28b69ad",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "SubnetId": "subnet-0d2570c9fd28b69ad",
        "State": "available",
        "VpcId": "vpc-0f0fb90e22f4d7bf4",
        "CidrBlock": "10.0.1.0/24",
        "AvailableIpAddressCount": 251,
        "AvailabilityZone": "ap-south-1a",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false
    }
}
```

```
C:\Users\Shivshankar>aws ec2 create-subnet --vpc-id vpc-0f0fb90e22f4d7bf4 --cidr-block 10.0.2.0/24 --av
ailability-zone ap-south-1a
{
    "Subnet": {
        "AvailabilityZoneId": "aps1-az1",
        "OwnerId": "715841363442",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:ap-south-1:715841363442:subnet/subnet-04c81a71ccd1ad9cf",
        "EnableDns64": false,
        "Ipv6Native": false,
        "PrivateDnsNameOptionsOnLaunch": {
            "HostnameType": "ip-name",
            "EnableResourceNameDnsARecord": false,
            "EnableResourceNameDnsAAAARecord": false
        },
        "SubnetId": "subnet-04c81a71ccd1ad9cf",
        "State": "available",
        "VpcId": "vpc-0f0fb90e22f4d7bf4",
        "CidrBlock": "10.0.2.0/24",
        "AvailableIpAddressCount": 251,
        "AvailabilityZone": "ap-south-1a",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false
    }
}
```

4. Create an Internet Gateway :

```
C:\Users\Shivshankar>aws ec2 create-internet-gateway
{
    "InternetGateway": {
        "Attachments": [],
        "InternetGatewayId": "igw-0fcab2df8663fea78",
        "OwnerId": "715841363442",
        "Tags": []
    }
}
```

5.Attach Internet Gateway to VPC

```
C:\Users\Shivshankar>aws ec2 attach-internet-gateway --internet-gateway-id igw-0fcab2df8663fea78 --vpc-
id vpc-0f0fb90e22f4d7bf4
```

6. Create a Route Table :

```
C:\Users\Shivshankar>aws ec2 create-route-table --vpc-id vpc-0f0fb90e22f4d7bf4
{
    "RouteTable": {
        "Associations": [],
        "PropagatingVgws": [],
        "RouteTableId": "rtb-0961bcc0c951215cd",
        "Routes": [
            {
                "DestinationCidrBlock": "10.0.0.0/16",
                "GatewayId": "local",
                "Origin": "CreateRouteTable",
                "State": "active"
            }
        ],
        "Tags": [],
        "VpcId": "vpc-0f0fb90e22f4d7bf4",
        "OwnerId": "715841363442"
    },
    "ClientToken": "2bb7ba15-0848-4479-be65-276cb6ff19ab"
}
```

7.Add route to Internet Gateway in Public Route Table

```
C:\Users\Shivshankar>aws ec2 create-route --route-table-id rtb-0961bcc0c951215cd --destination-cidr-blo
ck 0.0.0.0/0 --gateway-id igw-0fcab2df8663fea78
{
    "Return": true
}
```

8.# Associate Route Table with Public Subnet

```
C:\Users\Shivshankar>aws ec2 associate-route-table --route-table-id rtb-0961bcc0c951215cd --subnet-id s
ubnet-0d2570c9fd28b69ad
{
    "AssociationId": "rtbassoc-0cd1c9b0633e57047",
    "AssociationState": {
        "State": "associated"
    }
}
```

9. Modify Public Subnet to Enable Auto-assign Public IP :

```
C:\Users\Shivshankar>aws ec2 modify-subnet-attribute --subnet-id subnet-0d2570c9fd28b69ad --map-public-
ip-on-launch
```

10.Establish security groups and include Inbound rules in them:

Create Security Groups

```
C:\Users\Shivshankar>aws ec2 create-security-group --group-name web-sg --description "Security group fo
r web application" --vpc-id vpc-0f0fb90e22f4d7bf4
{
    "GroupId": "sg-032555aa1293ccb7f"
}
```

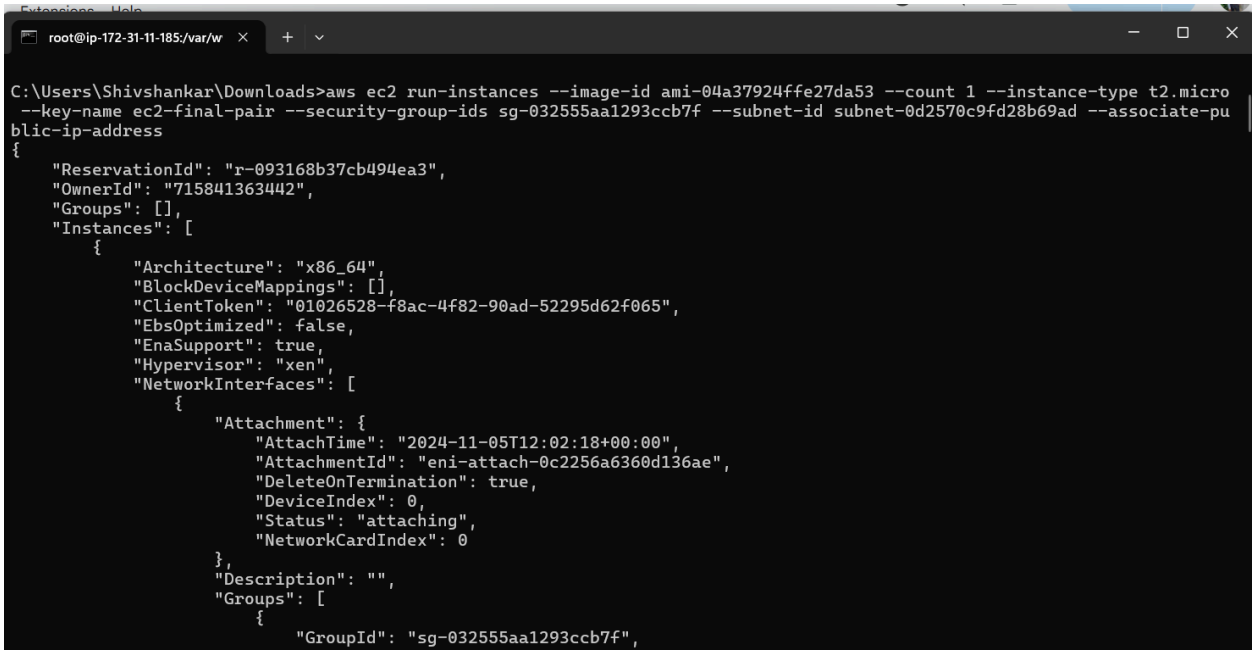
 # Add inbound rule for HTTP (port 80)

```
C:\Users\Shivshankar>aws ec2 authorize-security-group-ingress --group-id sg-032555aa1293ccb7f --protoco
l tcp --port 80 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0ddec66269c1aae0a",
            "GroupId": "sg-032555aa1293ccb7f",
            "GroupOwnerId": "715841363442",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

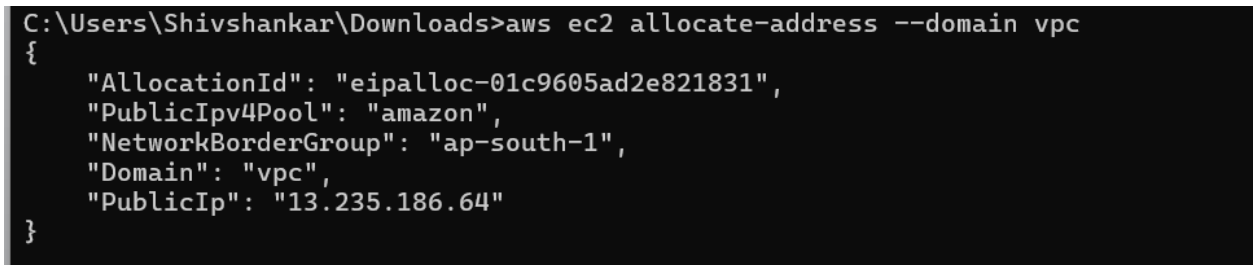
 # Add inbound rule for SSH (port 22)

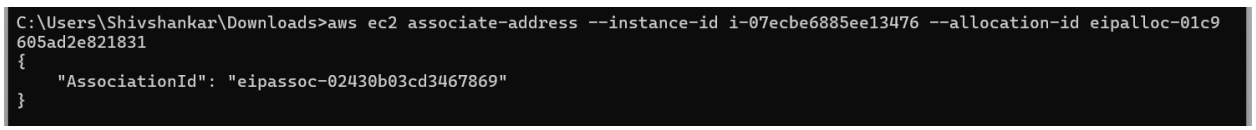
```
C:\Users\Shivshankar>aws ec2 authorize-security-group-ingress --group-id sg-032555aa1293ccb7f --protoco
l tcp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0a4af010711c2d621",
            "GroupId": "sg-032555aa1293ccb7f",
            "GroupOwnerId": "715841363442",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

11. Launch an EC2 Instance :

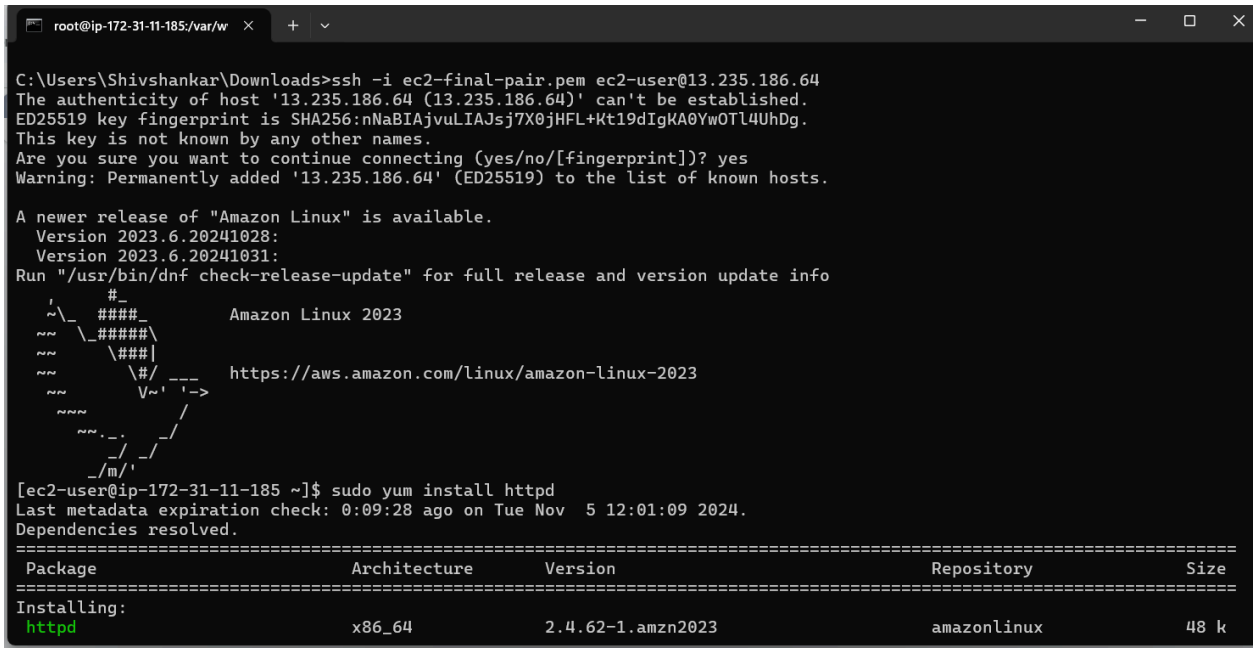


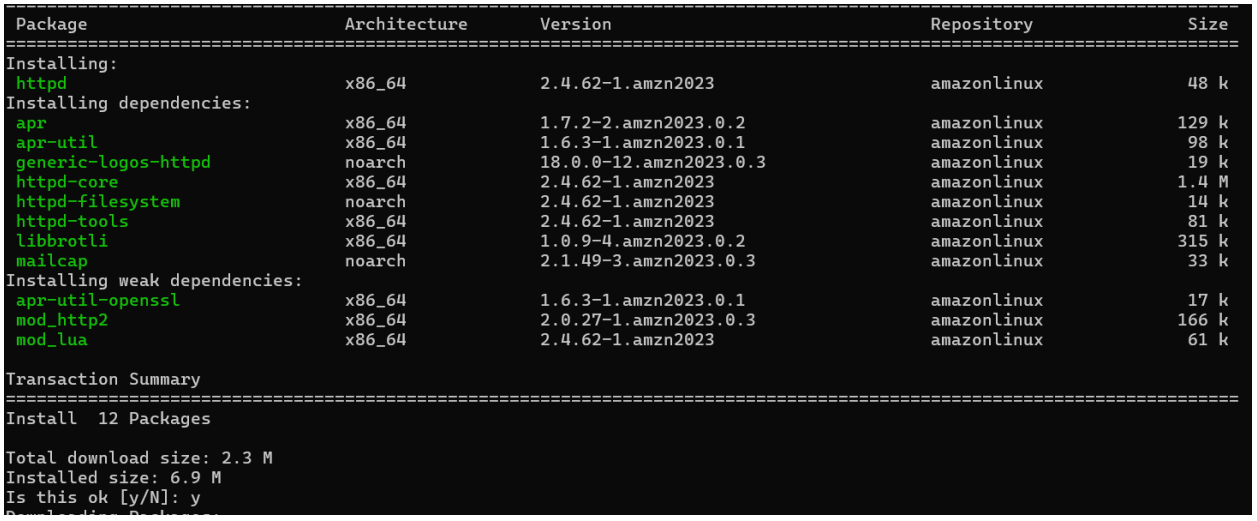
12. Allocate and Associate an Elastic IP :

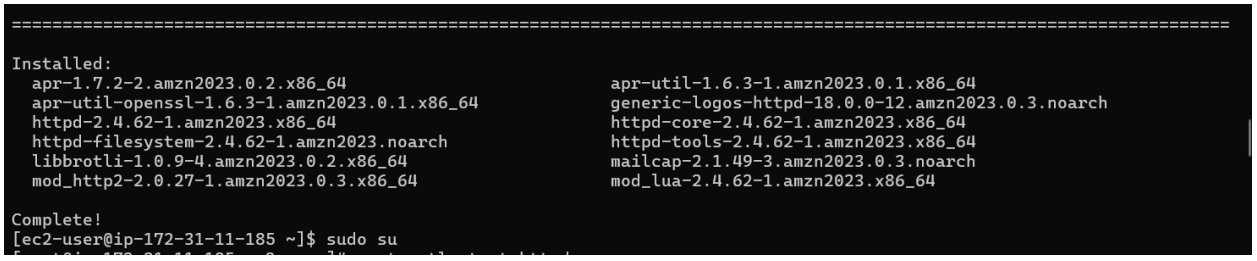




13.Connect To EC2 Instance :



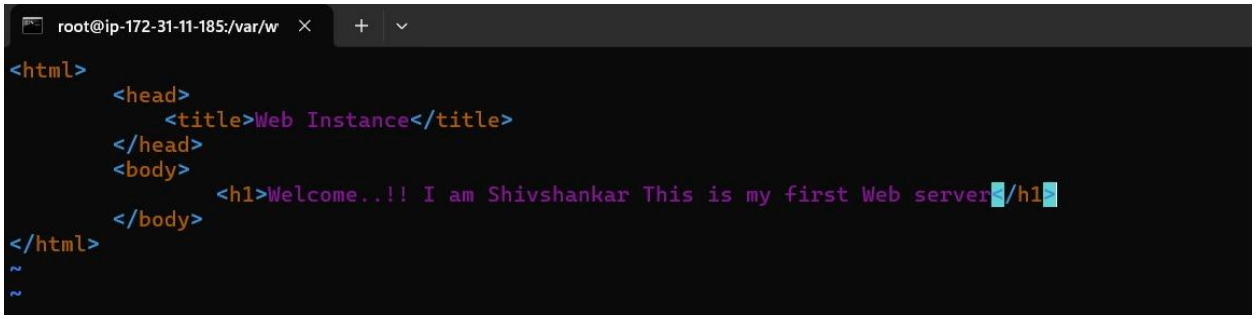




14.Web server

```
[ec2-user@ip-172-31-11-185 ~]$ sudo su
[root@ip-172-31-11-185 ec2-user]# systemctl start httpd
[root@ip-172-31-11-185 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Tue 2024-11-05 12:11:59 UTC; 18s ago
       Docs: man:httpd.service(8)
   Main PID: 25974 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 1112)
     Memory: 12.9M
        CPU: 63ms
     CGroup: /system.slice/httpd.service
             ├─25974 /usr/sbin/httpd -DFOREGROUND
             ├─25975 /usr/sbin/httpd -DFOREGROUND
             ├─25976 /usr/sbin/httpd -DFOREGROUND
             ├─25977 /usr/sbin/httpd -DFOREGROUND
             └─25978 /usr/sbin/httpd -DFOREGROUND

Nov 05 12:11:59 ip-172-31-11-185.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Serve>
Nov 05 12:11:59 ip-172-31-11-185.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Nov 05 12:11:59 ip-172-31-11-185.ap-south-1.compute.internal httpd[25974]: Server configured, listening on: port 80
```



Welcome..!! I am Shivshankar This is my first Web server