AWS Hand On

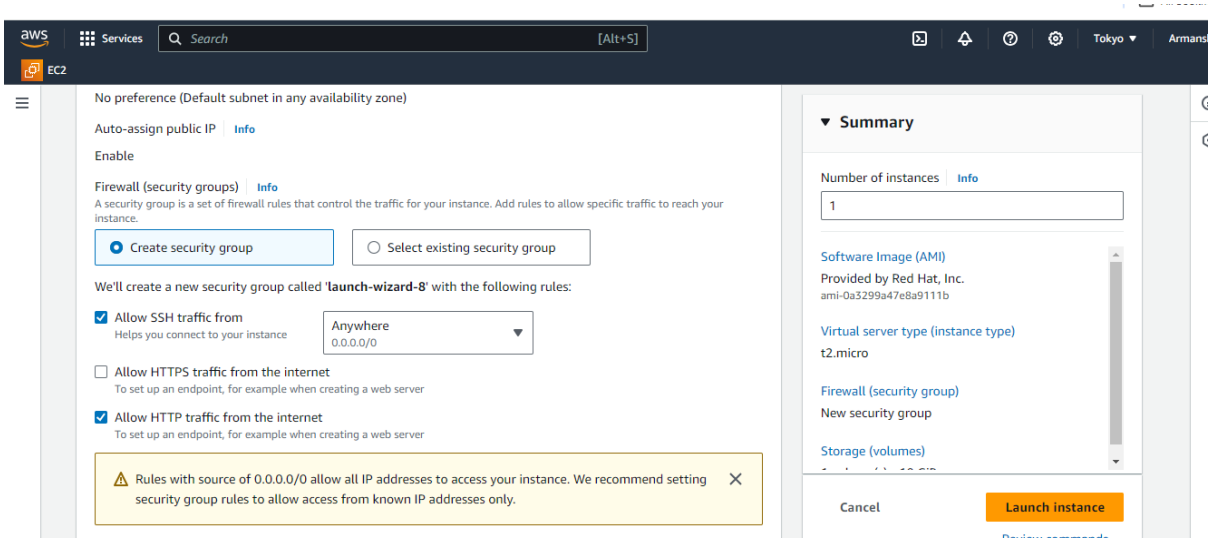
(ON CONSOLE)

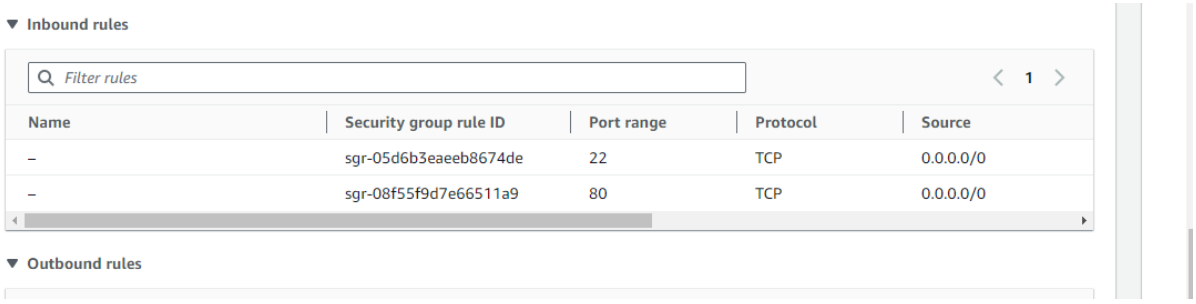
Q 01.

1. Create Security Group:

- Create one security group for the web server.

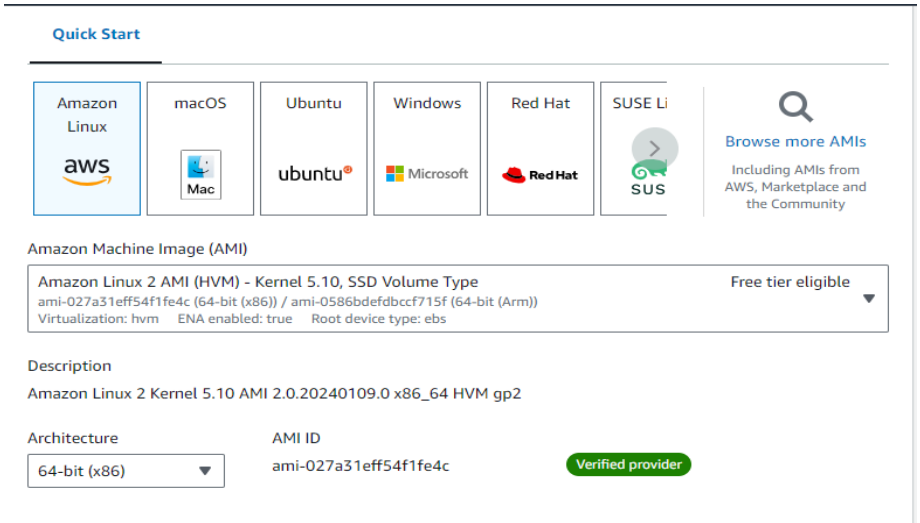
- Configure inbound rules for the web server security group to allow HTTP traffic (port 80) and SSH traffic (port 22) from any source.

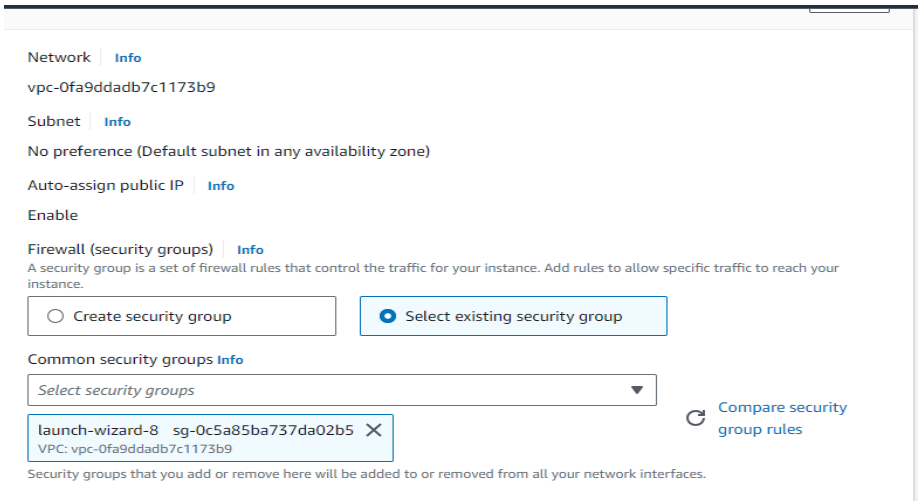


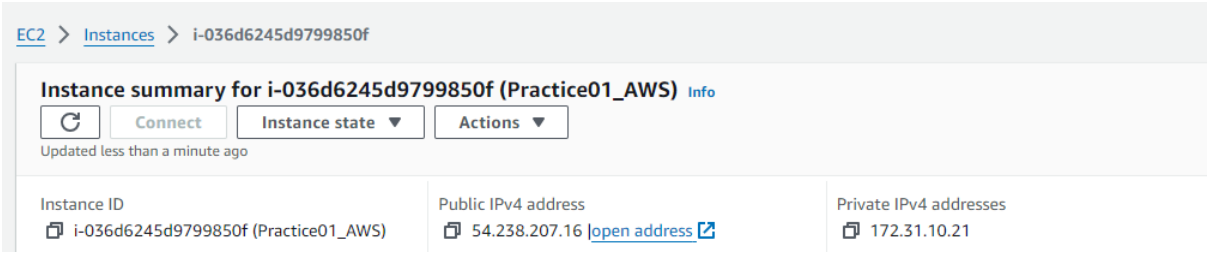


2. Launch EC2 Instance:

- Launch an EC2 instance for the web server using Amazon Linux 2 AMI.
  - Associate the web server security group created earlier with this instance.
  - Use an appropriate instance type for a web server.
  - Ensure the instance has a public IP address.

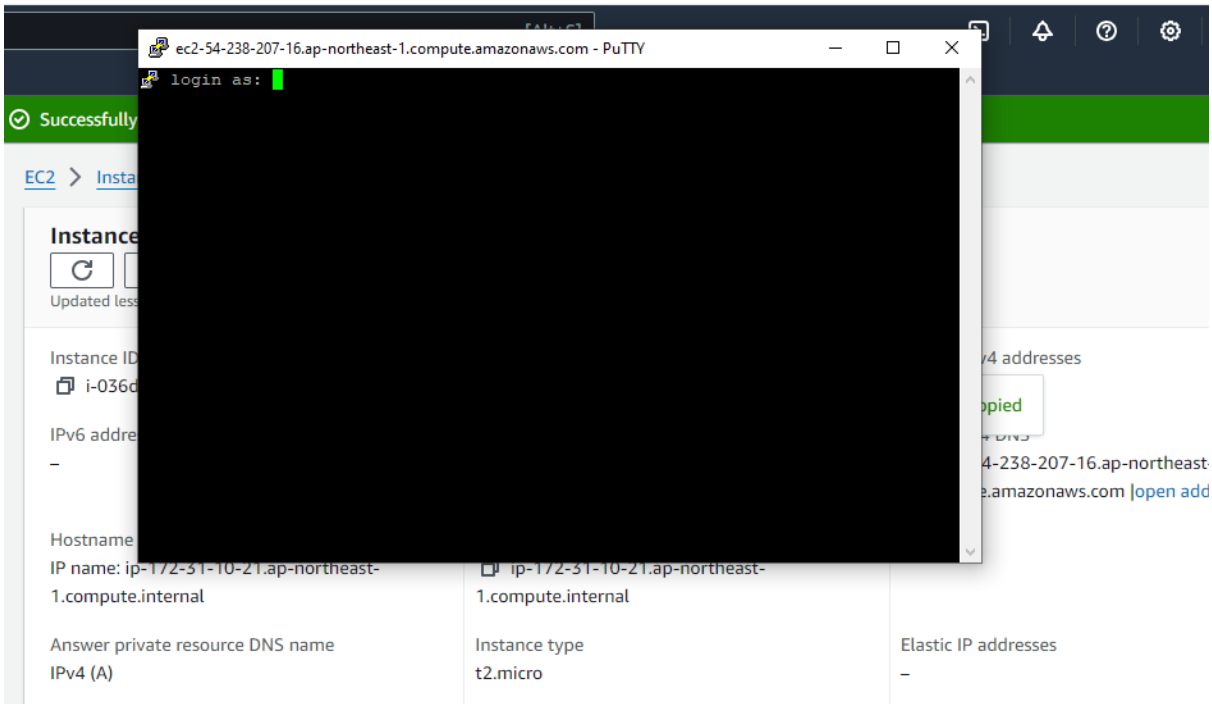






SSH Access:

- Generate an SSH key pair for secure access to the instances.
- Configure the web server instance to accept SSH connections using the generated key pair.
- Attempt to SSH into the web server instance to verify successful access.



Web Application Setup:

- Install a web server (e.g., Apache or Nginx) on the web server instance.
- Create a simple HTML page to confirm the web server is working.
- Test accessing the web server's public IP address in a web browser.

[root@ip-172-31-10-21 ~]# yum install httpd

[root@ip-172-31-10-21 html]# systemctl status httpd

● httpd.service - The Apache HTTP Server

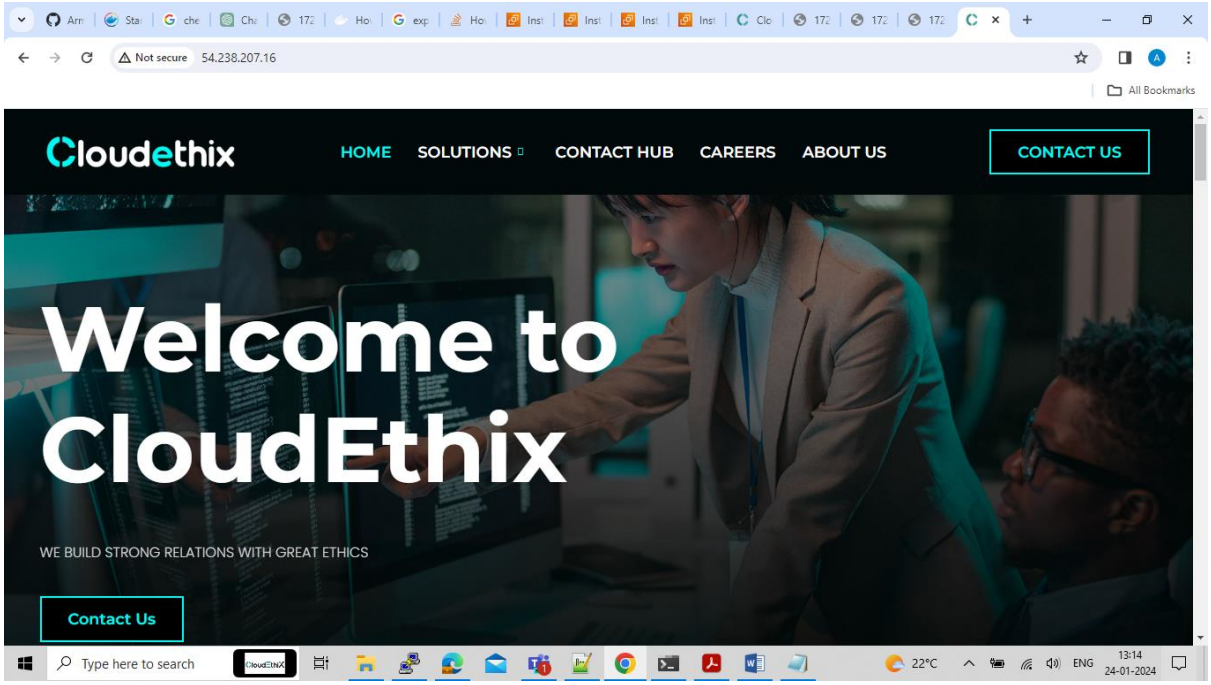
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)

Active: active (running) since Wed 2024-01-24 07:33:30 UTC; 14s ago

Docs: man:httpd.service(8)

Main PID: 3435 (httpd)

Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec



(ON CLI)

Q 02.

1. Create Security Group for Web Server Using AWS CLI:

- Use the AWS CLI to create a security group for the web server.
- Configure inbound rules to allow HTTP traffic (port 80) and SSH traffic (port 22) from any source.

```
root@DESKTOP-0ANCI6F:~ (hotfix)# aws ec2 create-security-group --description AWS3 --group-name
sec_grp
{
    "GroupId": "sg-0495fd09c1382b195"
}
root@DESKTOP-0ANCI6F:~ (hotfix)# aws ec2 authorize-security-group-ingress \
    --group-id sg-0495fd09c1382b195 \
    --protocol tcp \
    --port 80 \
    --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-03d7f4234fc556e13",
            "GroupId": "sg-0495fd09c1382b195",
            "GroupOwnerId": "010514484831",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
```

```
}
root@DESKTOP-0ANCI6F:~ (hotfix)# aws ec2 authorize-security-group-ingress \
    --group-id sg-0495fd09c1382b195 \
    --protocol tcp \
    --port 22 \
    --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-005c6ebf97ce844bf",
            "GroupId": "sg-0495fd09c1382b195",
            "GroupOwnerId": "010514484831",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

2. Launch EC2 Instance for Web Server Using AWS CLI:
- Use the AWS CLI to launch an EC2 instance for the web server using Amazon Linux 2 AMI.
- Associate the security group created earlier with this instance.
- Use an appropriate instance type for a web server.
- Ensure the instance has a public IP address.

```
root@DESKTOP-0ANCI6F:~ (hotfix)# aws ec2 run-instances --image-id ami-027a31eff54f1fe4c --instance-type t2.micro --key-name 10lpa --security-group-ids sg-0c5a85ba737da02b5 --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Practice02_AWS}]'
```


```
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-027a31eff54f1fe4c",
            "InstanceId": "i-06ed33997c60a4c68",
            "InstanceType": "t2.micro",
            "KeyName": "10lpa",
            "LaunchTime": "2024-01-24T09:46:34.000Z",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "ap-northeast-1c",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-14-105.ap-northeast-1.compute.internal",
            "PrivateIpAddress": "172.31.14.105",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
```

3. SSH Access Using AWS CLI:

- Use the AWS CLI to generate an SSH key pair for secure access to the web server instance.
- Configure the web server instance to accept SSH connections using the generated key pair.
- Use the AWS CLI to attempt to SSH into the web server instance to verify successful access.

```
root@DESKTOP-1RT156R:/mnt/c/Users/ADMIN/Downloads# aws ec2 describe-instances --instance-ids i-0909bd3145668dde8 --query 'Reservations[0].Instances[0].PublicIpAddress' --output text
50.17.77.247
root@DESKTOP-1RT156R:/mnt/c/Users/ADMIN/Downloads# chmod 400 key2.pem
root@DESKTOP-1RT156R:/mnt/c/Users/ADMIN/Downloads# ssh -i key2.pem ec2-user@50.17.77.247
 ,     #_
 ~\_  ####_        Amazon Linux 2023
~~  \_#####\
~~     \###|
~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
 ~~       V~' '->
  ~~~         /
    ~~._.   _/
       _/ _/
     _/m/'
```

4. Web Application Setup Using AWS CLI:

- Use the AWS CLI to install a web server (e.g., Apache or Nginx) on the web server instance.
- Create a simple HTML page using the AWS CLI to confirm the web server is working.
- Use the AWS CLI to test accessing the web server's public IP address in a web browser.

```
root@DESKTOP-0ANCI6F: # aws ec2 run-instances --image-id ami-0e9107ed11be76fde --key-name star1 --instance-type t2.micro --security-group-ids sg-0c7227cd683f4213d --associate-public-ip-address --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Ec2_Instance}]' --user-data file://user-data.sh
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0e9107ed11be76fde",
            "InstanceId": "i-07f71bd5cedacef5f",
            "InstanceType": "t2.micro",
            "KeyName": "star1",
            "LaunchTime": "2024-01-20T14:23:10.000Z",
```

"Monitoring": {
  "State": "disabled"
},
"Placement": {
  "AvailabilityZone": "us-east-1a",
  "GroupName": "",
  "Tenancy": "default"
},
"PrivateDnsName": "ip-172-31-19-51.ec2.internal",
"PrivateIpAddress": "172.31.19.51",
"ProductCodes": [],
"PublicDnsName": "",
"State": {
  "Code": 0,
  "Name": "pending"
},
"StateTransitionReason": "",
"SubnetId": "subnet-046296ed3451035b6",
"VpcId": "vpc-0e40e229396d8047f",
"Architecture": "x86_64",
"BlockDeviceMappings": [],
"ClientToken": "6e513087-8d75-420b-81ff-459a776cdd6e",
"EbsOptimized": false,
"EnaSupport": true,
"Hypervisor": "xen",
"NetworkInterfaces": [
  {
    "Attachment": {
      "AttachTime": "2024-01-20T14:23:10.000Z",
      "AttachmentId": "eni-attach-00e7e365c3f102796",
      "DeleteOnTermination": true,
      "DeviceIndex": 0,
      "Status": "attaching",
      "NetworkCardIndex": 0
    },
    "Description": "",
    "Groups": [
      {
        "GroupName": "Mysg",
        "GroupId": "sg-0c7227cd683f4213d"
      }
    ],
    "Ipv6Addresses": [],
    "MacAddress": "0a:e2:bc:f8:1d:91",
    "NetworkInterfaceId": "eni-0e0d0b80a3d3af1bc",
    "OwnerId": "842313196830",
    "PrivateDnsName": "ip-172-31-19-51.ec2.internal",
    "PrivateIpAddress": "172.31.19.51",

```
        "PrivateIpAddresses": [
          {
            "Primary": true,
            "PrivateDnsName": "ip-172-31-19-51.ec2.internal",
            "PrivateIpAddress": "172.31.19.51"
          }
        ],
        "SourceDestCheck": true,
        "Status": "in-use",
        "SubnetId": "subnet-046296ed3451035b6",
        "VpcId": "vpc-0e40e229396d8047f",
        "InterfaceType": "interface"
    }
  ],
  "RootDeviceName": "/dev/xvda",
  "RootDeviceType": "ebs",
  "SecurityGroups": [
    {
      "GroupName": "Mysg",
      "GroupId": "sg-0c7227cd683f4213d"
    }
  ],
  "SourceDestCheck": true,
  "StateReason": {
    "Code": "pending",
    "Message": "pending"
  },
  "Tags": [
    {
      "Key": "Name",
      "Value": "Ec2_Instance"
    }
  ],
  "VirtualizationType": "hvm",
  "CpuOptions": {
    "CoreCount": 1,
    "ThreadsPerCore": 1
  },
  "CapacityReservationSpecification": {
    "CapacityReservationPreference": "open"
  },
  "MetadataOptions": {
    "State": "pending",
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 2,
    "HttpEndpoint": "enabled",
    "HttpProtocolIpv6": "disabled",
    "InstanceMetadataTags": "disabled"
```

```
            },
            "EnclaveOptions": {
                "Enabled": false
            },
            "BootMode": "uefi-preferred",
            "PrivateDnsNameOptions": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            }
        }
    ],
    "OwnerId": "842313196830",
    "ReservationId": "r-07052f2493a5551d3"
}
root@DESKTOP-0ANCI6F:# aws ec2 describe-instances  --
QUERY^CReservations[0].Instances[0].PublicIpAddress' --output text
root@DESKTOP-0ANCI6F:# aws ec2 describe-instances --instance-ids i-07f71bd5cedacef5f --query
'Reservations[0].Instances[0].PublicIpAddress' --output text
34.230.70.246
root@DESKTOP-0ANCI6F:# ssh -i star1.pem ec2-user@34.230.70.246
The authenticity of host '34.230.70.246 (34.230.70.246)' can't be established.
ED25519 key fingerprint is SHA256:GEy+A564805CGzanNzc/pn76KcRYMTlB7417ofWafxg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.230.70.246' (ED25519) to the list of known hosts.
  ,     #_
  ~\_  ####_       Amazon Linux 2023
 ~~  \_#####\
 ~~     \###|
 ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~' '->
   ~~~         /
     ~~._.   _/
        _/ _/
      _/m/'
[ec2-user@ip-172-31-19-51 ~]$ netstat -tunpl
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
udp        0      0 127.0.0.1:323           0.0.0.0:*                           -
udp        0      0 172.31.19.51:68         0.0.0.0:*                           -
udp6       0      0 ::1:323                 :::*                                -
udp6       0      0 fe80::8e2:bcff:fef8:546 :::*                                -
```

```
[ec2-user@ip-172-31-19-51 ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2024-01-20 14:24:07 UTC; 35s ago
     Docs: man:httpd.service(8)
 Main PID: 3572 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
    Tasks: 177 (limit: 1114)
   Memory: 13.1M
      CPU: 75ms
   CGroup: /system.slice/httpd.service
           ├─3572 /usr/sbin/httpd -DFOREGROUND
           ├─3596 /usr/sbin/httpd -DFOREGROUND
           ├─3598 /usr/sbin/httpd -DFOREGROUND
           ├─3599 /usr/sbin/httpd -DFOREGROUND
           └─3600 /usr/sbin/httpd -DFOREGROUND
```