

CREATE THREE VPCs IN THREE DIFFERENT REGIONS AND CONNECT THE VPCs USING TRANSIT GATEWAY

➤ What is Amazon VPC?

With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data centre, with the benefits of using the scalable infrastructure of AWS.

➤ What is Transit Gateway?

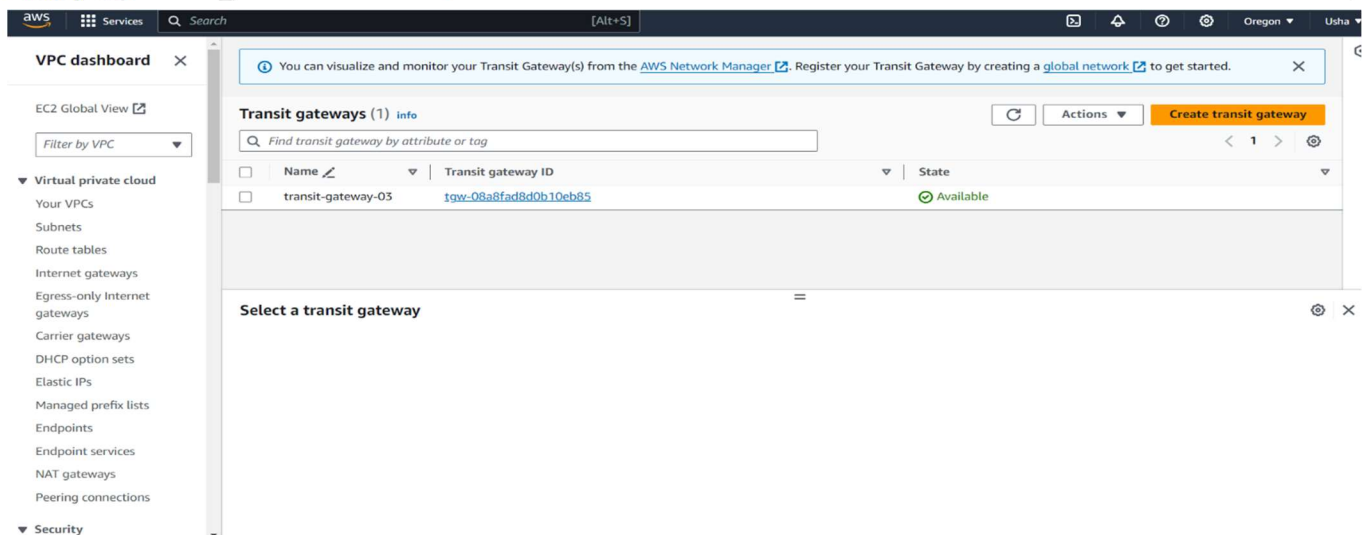
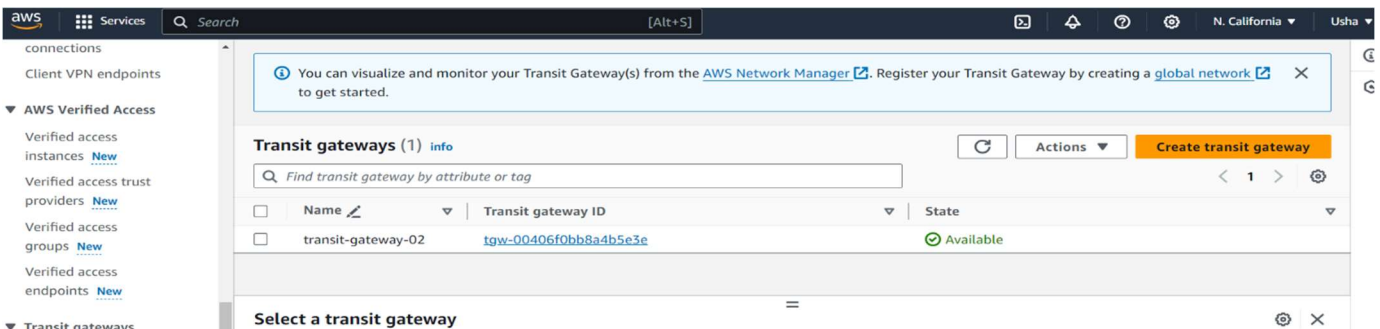
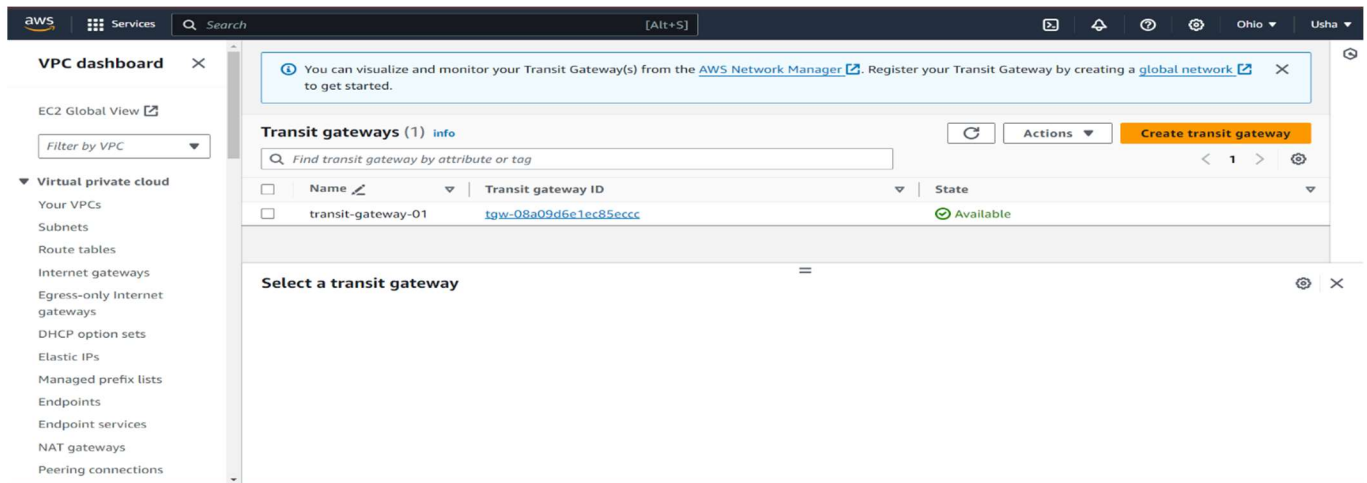
A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPCs) and on-premises networks. As your cloud infrastructure expands globally, inter-region peering connects transit gateways together using the AWS global infrastructure.

➤ Create VPCs in THREE DIFFERENT REGIONS (Ohio California Oregon):

1. Log in to your AWS account & select Ohio region & do search for VPC in the search box.
2. Click on create VPC & select VPC AND MORE, go down click on create VPC.
3. After that choose CALIFORNIA & OREGON and create VPCs in those regions.
4. Follow the steps mentioned in below snapshots.

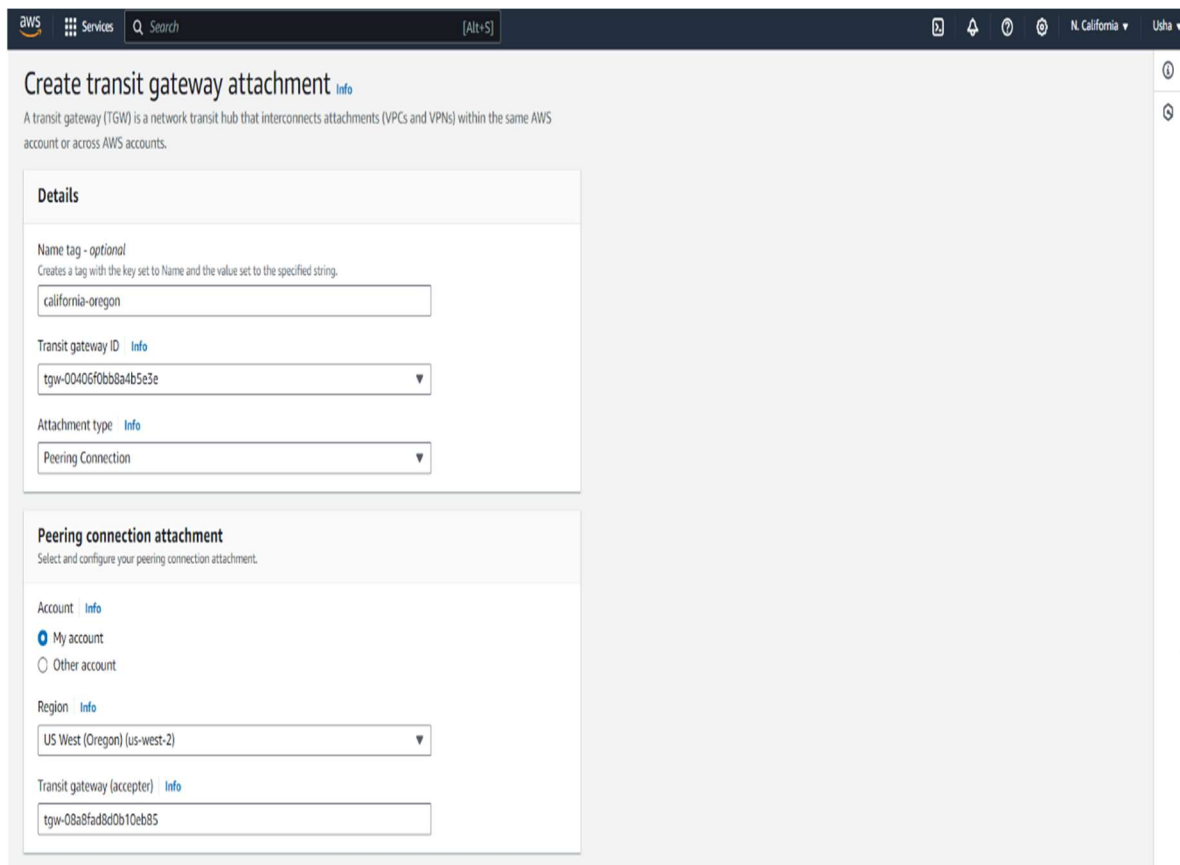
➤ Create Transit Gateway in three regions:

1. Go to transit gateway and select “Create Transit Gateway”.
2. Now create transit gateway in another two regions also.
3. Snapshots of transit gateway are attached below.



➤ Create Transit Gateway Attachment in THREE Regions:

1. Go to transit gateway attachment & Create “Transit Gateway Attachment” & attach to transit gateway.
2. while creating transit gateway attachment for single region, select attachment type attach VPC & for two or more regions give PEERING CONNECTION.
3. After sending a request from one region to another region, you must accept the transit gateway attachment request then only it will be available.
4. some snapshots attached below.



Create transit gateway attachment [Info](#)

A transit gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same AWS account or across AWS accounts.

Details

Name tag - optional
Creates a tag with the key set to Name and the value set to the specified string.

california-oregon

Transit gateway ID [Info](#)

tgw-00406f0bb8a4b5e3e

Attachment type [Info](#)

Peering Connection

Peering connection attachment
Select and configure your peering connection attachment.

Account [Info](#)

☒ My account
☐ Other account

Region [Info](#)

US West (Oregon) (us-west-2)

Transit gateway (accepter) [Info](#)

tgw-08a8fad8d0b10eb85

VPC dashboard X

EC2 Global View [↗](#)

Filter by VPC ▾

▼ Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only Internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- Endpoints
- Endpoint services
- NAT gateways
- Peering connections

You can visualize and monitor your Transit Gateway(s) from the [AWS Network Manager](#). Register your Transit Gateway by creating a [global network](#) to get started.

Transit gateway attachments (1/2) info

Find transit gateway attachment by attribute or tag

Name	Transit gateway attachment ID	Transit gateway ID	State
<input checked="" type="checkbox"/> tgw-attach-0929d42c2112cde45	tgw-attach-0929d42c2112cde45	tgw-08a8fad8d0b10eb85	Pending
<input type="checkbox"/> t-g-w-a-03	tgw-attach-0c143b305d1f8f84d	tgw-08a8fad8d0b10eb85	Available

Actions ▴

- Create transit gateway attachment
- Create flow log
- Manage tags
- Accept transit gateway attachment
- Reject transit gateway attachment
- Delete transit gateway attachment

Transit gateway attachment: tgw-attach-0929d42c2112cde45

Details | Flow logs | Tags

Details

Transit gateway attachment ID tgw-attach-0929d42c2112cde45	Requester ID tgw-00406f0bb8a4b5e3e	Acceptor ID tgw-08a8fad8d0b10eb85	State Pending Acceptance
Requester region N. California (us-west-1)	Acceptor region Oregon (us-west-2)	Resource type Peering	Requester owner ID 010928185144

VPC dashboard X

EC2 Global View [↗](#)

Filter by VPC ▾

▼ Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only Internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- Endpoints
- Endpoint services
- NAT gateways
- Peering connections

You can visualize and monitor your Transit Gateway(s) from the [AWS Network Manager](#). Register your Transit Gateway by creating a [global network](#) to get started.

Transit gateway attachments (1/2) info

Find transit gateway attachment by attribute or tag

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID
<input checked="" type="checkbox"/> tgw-attach-0929d42c2112cde45	tgw-attach-0929d42c2112cde45	tgw-08a8fad8d0b10eb85	Pending Acceptance	Peering	tgw-00406f0bb8a4b5e3e
<input type="checkbox"/> t-g-w-a-03	tgw-attach-0c143b305d1f8f84d	tgw-08a8fad8d0b10eb85	Available	VPC	vpc-c02c528c8b16ca185

Accept

Are you sure that you want to accept this transit gateway peering attachment tgw-attach-0929d42c2112cde45?

Cancel Accept

Transit gateway attachment: tgw-attach-0929d42c2112cde45

Details | Flow logs | Tags

Details

Transit gateway attachment ID tgw-attach-0929d42c2112cde45	Requester ID tgw-00406f0bb8a4b5e3e	Acceptor ID tgw-08a8fad8d0b10eb85	State Pending Acceptance
Requester region N. California (us-west-1)	Acceptor region Oregon (us-west-2)	Resource type Peering	Requester owner ID 010928185144

5. Below snapshots shows that transit gateway attachment from region – region.
i.e., Ohio-California, California-Oregon, Oregon-Ohio.

The first screenshot shows the Ohio region with three Transit Gateway attachments:

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID	Association route table ID
ohio-california	tgw-attach-0b9bcd37c573bdda9	tgw-08a09d6e1ec85eccc	Available	Peering	tgw-08a8fad8d0b10eb85	tgw-rtb-0d0103cf42
t-g-w-a-01	tgw-attach-011ee38ebd892f78f	tgw-08a09d6e1ec85eccc	Available	VPC	vpc-049494c3cdd9c4256	tgw-rtb-0d0103cf42

The second screenshot shows the N. California region with three Transit Gateway attachments:

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID	Association route table ID
california-oregon	tgw-attach-0929d42c2112cde45	tgw-00406f0bb8a4b5e3e	Available	Peering	tgw-08a8fad8d0b10eb85	tgw-rtb-0417
t-g-w-a-02	tgw-attach-0d1c0e24426445c51	tgw-00406f0bb8a4b5e3e	Available	Peering	tgw-08a09d6e1ec85eccc	tgw-rtb-0417
t-g-w-a-03	tgw-attach-0c0fc3bd1e52737b9	tgw-00406f0bb8a4b5e3e	Available	VPC	vpc-0d4ea057b4a3c896a	tgw-rtb-0417

The third screenshot shows the Oregon region with three Transit Gateway attachments:

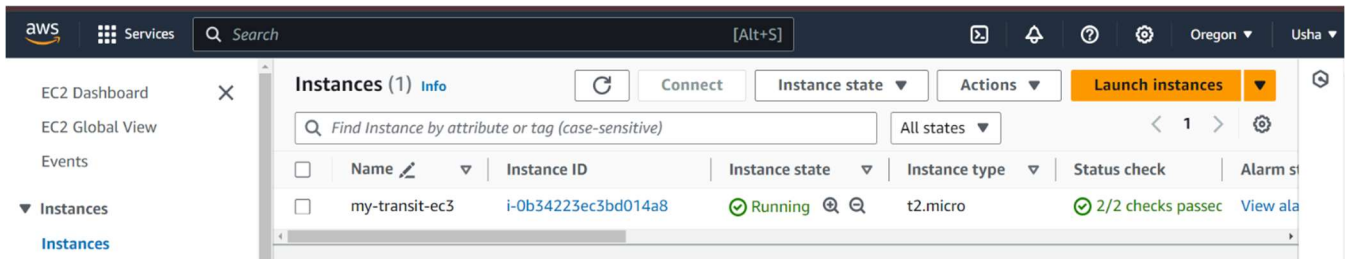
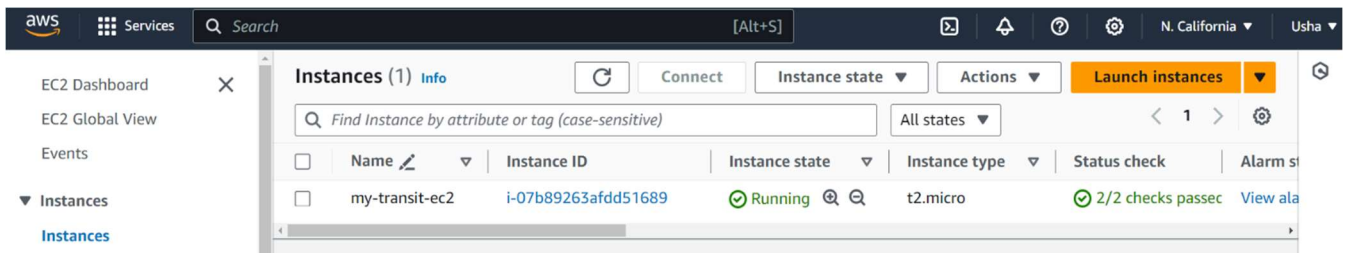
Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID	Association route table ID
oregon-ohio	tgw-attach-0b9bcd37c573bdda9	tgw-08a8fad8d0b10eb85	Available	Peering	tgw-00406f0bb8a4b5e3e	tgw-rtb-0417
t-g-w-a-02	tgw-attach-0d1c0e24426445c51	tgw-00406f0bb8a4b5e3e	Available	Peering	tgw-08a09d6e1ec85eccc	tgw-rtb-0417
t-g-w-a-03	tgw-attach-0c0fc3bd1e52737b9	tgw-00406f0bb8a4b5e3e	Available	VPC	vpc-0d4ea057b4a3c896a	tgw-rtb-0417

➤ Create EC2 instance:

1. create three EC2 Instance for three VPCs.
2. Go to instance –launch instance – create key pair–network(edit)–select security group – launch instance.
3. some EC2 snapshots are attached below.

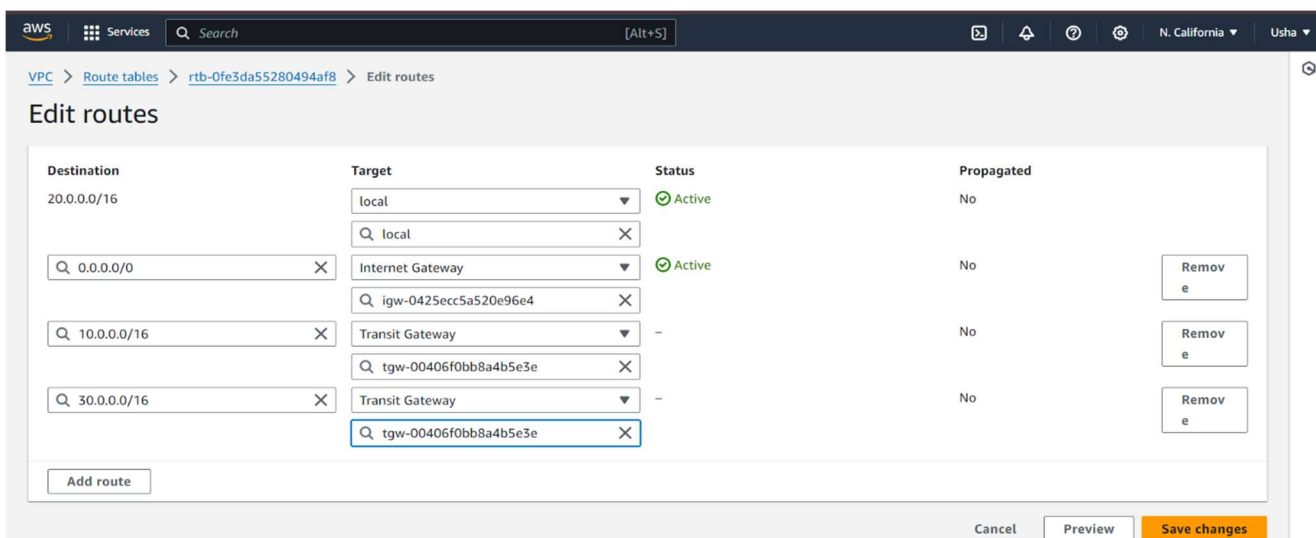
The screenshot shows the EC2 dashboard with one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
my-transit-ec1	i-0fa4ccd0077d54c9f	Running	t2.micro	Initializing	View alarms



4. After creating EC2 Instances, go to route tables & edit the routes & save changes.

5. Some snapshots of route tables are given below.



aws Services Search [Alt+S] Ohio Usha

VPC > Route tables > rtb-05052b19ebe07bc2f > Edit routes

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local	Active	No	
Q 20.0.0.0/16 X	Transit Gateway	Active	No	Remove
Q tgw-08a09d6e1ec85eccc X	Transit Gateway	Active	No	Remove
Q tgw-08a09d6e1ec85eccc X	Transit Gateway	Active	No	Remove
Q 0.0.0.0/0 X	Internet Gateway	Active	No	Remove
	igw-0fe43c27a488dcd57			

Add route

Cancel Preview Save changes

aws Services Search [Alt+S] Oregon Usha

VPC > Route tables > rtb-038000a7678abac7f > Edit routes

Edit routes

Destination	Target	Status	Propagated	
30.0.0.0/16	local	Active	No	
Q 0.0.0.0/0 X	Internet Gateway	Active	No	Remove
Q igw-0d17529dcbdcc2575 X	Internet Gateway			
Q 10.0.0.0/16 X	Transit Gateway	-	No	Remove
Q tgw-08a8fad8d0b10eb85 X	Transit Gateway	-	No	Remove
Q 20.0.0.0/16 X	Transit Gateway	-	No	Remove
	tgw-08a8fad8d0b10eb85			

Add route

Cancel Preview Save changes

6. Goto the EC2 Instance click on connect – copy The SSH and paste it in git bash & connect.
7. Directly connect to the server through EC2 Instance.


```
Verifying : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 7/7

Installed:
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  gperftools-libe-2.9.1-1.amzn2023.0.3.x86_64  libunwind-1.4.0-5.amzn2023.0.2.x86_64
nginx-1.1.24.0-1.amzn2023.0.2.x86_64  nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64  nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-10-0-9-114 ~]# cd /usr/share/nginx/html
[root@ip-10-0-9-114 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-10-0-9-114 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-10-0-9-114 html]# vi index.html
[root@ip-10-0-9-114 html]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: inactive (dead)
[root@ip-10-0-9-114 html]# systemctl restart nginx
[root@ip-10-0-9-114 html]# systemctl start nginx
[root@ip-10-0-9-114 html]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: active (running) since Fri 2024-08-16 05:50:18 UTC; 30s ago
     Process: 26141 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 26142 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 26143 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 26144 (nginx)
      Tasks: 2 (limit: 1112)
     Memory: 2.2M
        CPU: 54ms
    CGroup: /system.slice/nginx.service
            └─26144 "nginx: master process /usr/sbin/nginx"
              └─26145 "nginx: worker process"

Aug 16 05:50:18 ip-10-0-9-114.us-east-2.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 16 05:50:18 ip-10-0-9-114.us-east-2.compute.internal nginx[26142]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 16 05:50:18 ip-10-0-9-114.us-east-2.compute.internal nginx[26142]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 16 05:50:18 ip-10-0-9-114.us-east-2.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-10-0-9-114 html]#
```

➤ OUTPUT:

```
0 upgraded, 0 newly installed, 0 to remove and 53 not upgraded.
root@ip-20-0-13-141:~# cd /var/www/html
root@ip-20-0-13-141:/var/www/html# ls
index.html
root@ip-20-0-13-141:/var/www/html# cat index.html

this is a california server
root@ip-20-0-13-141:/var/www/html# ls
index.html
root@ip-20-0-13-141:/var/www/html# rm index.html
root@ip-20-0-13-141:/var/www/html# vi index.html
root@ip-20-0-13-141:/var/www/html# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-08-16 05:59:35 UTC; 36min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2916 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.6M (peak: 5.8M)
        CPU: 136ms
    CGroup: /system.slice/apache2.service
            └─2916 /usr/sbin/apache2 -k start
              └─2919 /usr/sbin/apache2 -k start
                └─2920 /usr/sbin/apache2 -k start

Aug 16 05:59:35 ip-20-0-13-141 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 16 05:59:35 ip-20-0-13-141 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-20-0-13-141:/var/www/html# curl 18.222.22.5:80
curl18.222.22.5:80: command not found
root@ip-20-0-13-141:/var/www/html# curl 18.222.22.5:80
this is ohio server
root@ip-20-0-13-141:/var/www/html# curl 13.57.31.138:80
this is california server
root@ip-20-0-13-141:/var/www/html# curl 54.190.237.120:80
this is a oregon server
root@ip-20-0-13-141:/var/www/html# exit
logout
ubuntu@ip-20-0-13-141:~$ exit
logout
```

```
aws Services Q Search [Alt+S] Oregon Usha

[root@ip-30-0-6-132 html]# systemctl status nginx
o nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: inactive (dead)
[root@ip-30-0-6-132 html]# systemctl start nginx
[root@ip-30-0-6-132 html]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: active (running) since Fri 2024-08-16 06:05:33 UTC; 5s ago
     Process: 26541 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 26542 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 26543 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Main PID: 26544 (nginx)
    Tasks: 2 (limit: 1112)
   Memory: 2.2M
      CPU: 55ms
   CGroup: /system.slice/nginx.service
           └─26544 "nginx: master process /usr/sbin/nginx"
             └─26545 "nginx: worker process"

Aug 16 06:05:33 ip-30-0-6-132.us-west-2.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 16 06:05:33 ip-30-0-6-132.us-west-2.compute.internal nginx[26542]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 16 06:05:33 ip-30-0-6-132.us-west-2.compute.internal nginx[26542]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 16 06:05:33 ip-30-0-6-132.us-west-2.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-30-0-6-132 html]# curl 13.57.31.138:80

this is a california server
[root@ip-30-0-6-132 html]# curl 54.190.237.120:80
this is a oregon server
[root@ip-30-0-6-132 html]# curl 18.222.22.5:80
this is ohio server
[root@ip-30-0-6-132 html]#
```