

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 have defined. This virtual network resembles a traditional network that you would 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 VPCs and on-premises networks. As your cloud infrastructure expands globally, inter region peering connects transit gateways together using the AWS global infrastructure.

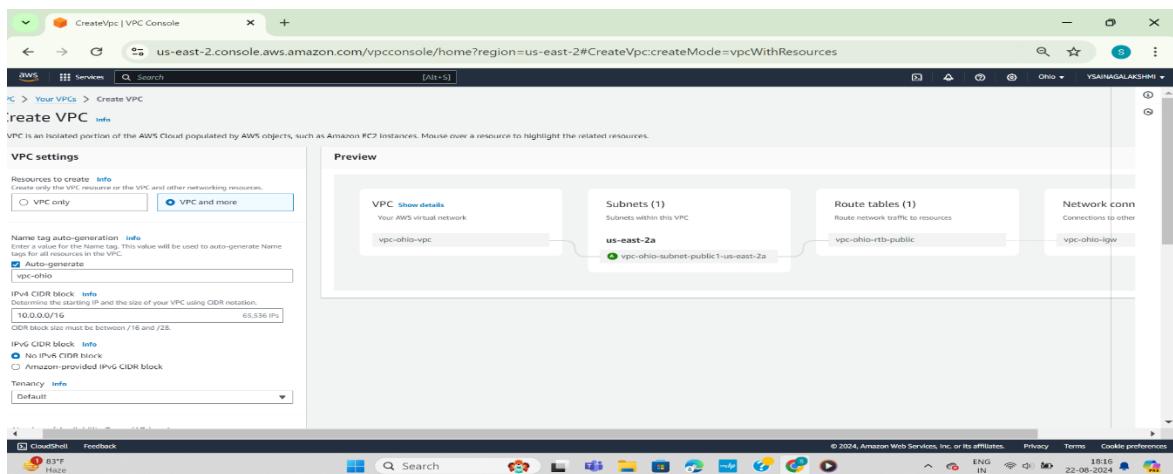
CREATE VPCs in THREE DIFFERENT REGIONS:

Regions – Ohio, California, Oregon.

STEPS:

1. Login to AWS Account & Select the regions (Ohio, California, Oregon).
2. Choose the **Ohio** region, Click on Create VPC, Select VPC and more and create VPC.
3. Do the same process after selecting the **California & Oregon** Regions.

OHIO:



VPC dashboard

Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
vpc-chilo-vpc	vpc-0b2eff7a6d2d2740e	Available	10.0.0.0/16	-	dopt-0fd655906aae6e	rtb-0909461cd24146ca
-	vpc-001e153b7e97be6d	Available	172.31.0.0/16	-	dopt-0fd655906aae6e	rtb-090e239af4a12905

Select a VPC above

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CALIFORNIA:

CreateVpc | VPC Console

us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#CreateVpc:createMode=vpcWithResources

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create **Info** Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation **Info** Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

Auto-generate
 vpc-california

IPv4 CIDR Block **Info** Determine the starting IP and the size of your VPC using CIDR notation.

10.50.0.0/16 65,536 IPs

IPv6 CIDR block **Info** No IPv6 CIDR block

Tenancy **Info**

Preview

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

vpc | VPC Console

us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#vpces:

VPC dashboard

Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set
vpc-california-vpc	vpc-021602f07502ddcc	Available	10.50.0.0/16	-	dopt-04904ac79b2c66e...
-	vpc-0950073eb0499ea11	Available	172.31.0.0/16	-	dopt-04904ac79b2c66e...

Select a VPC above

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

OREGON:

VPC settings

Resources to create: **VPC only** (selected) **VPC and more**

Name tag auto-generation: **Auto-generate** (selected)

IPv4 CIDR block: **10.60.0.0/16** (65,536 IPs)

IPv6 CIDR block: **No IPv6 CIDR block** (selected)

Tenancy: **VPC only**

Preview

VPC: **vpc-oregon-vpc** Your AWS virtual network

Subnets (1): **us-west-2a** Subnets within this VPC

Route tables (1): **vpc-oregon-rtb-public** Route network traffic to resources

Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set
vpc-0f1c7de7a22bba90f	vpc-0f1c7de7a22bba90f	Available	10.60.0.0/16	-	dopt-00129a59a779...
vpc-oregon-vpc	vpc-0f1c7de7a22bba90f	Available	10.60.0.0/16	-	dopt-00129a59a779...

Select a VPC above

Region: Oregon

4.Create EC2 instances for 3 regions.

- Launch instance – create key pair - create security group - add security group rule - launch instance.

OHIO:

Name and tags

Name: **ec2-ohio**

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Quick Start

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

Virtualization: **Amazon VPC** (Default) **aws** (Default)

Virtualization: **Amazon VPC** (Default) **aws** (Default)

Description: Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support.

Summary

Number of instances: **1**

Software Image (AMI): **Amazon Linux 2023 AMI 2023.5.2... (read more)**

Virtual server type (instance type): **t2.micro**

Firewall (security group): **New security group**

Storage (volumes): **1 volume(s) - 8 GB**

Launch instance

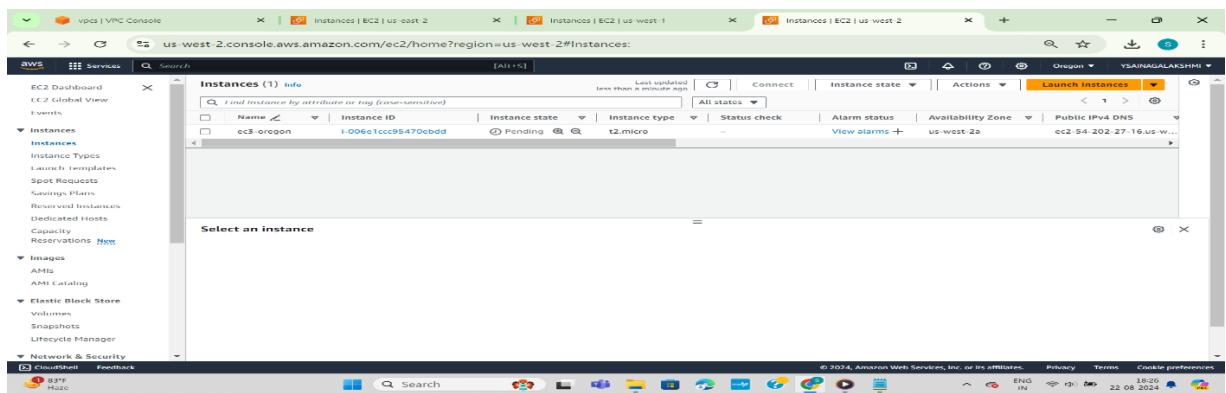
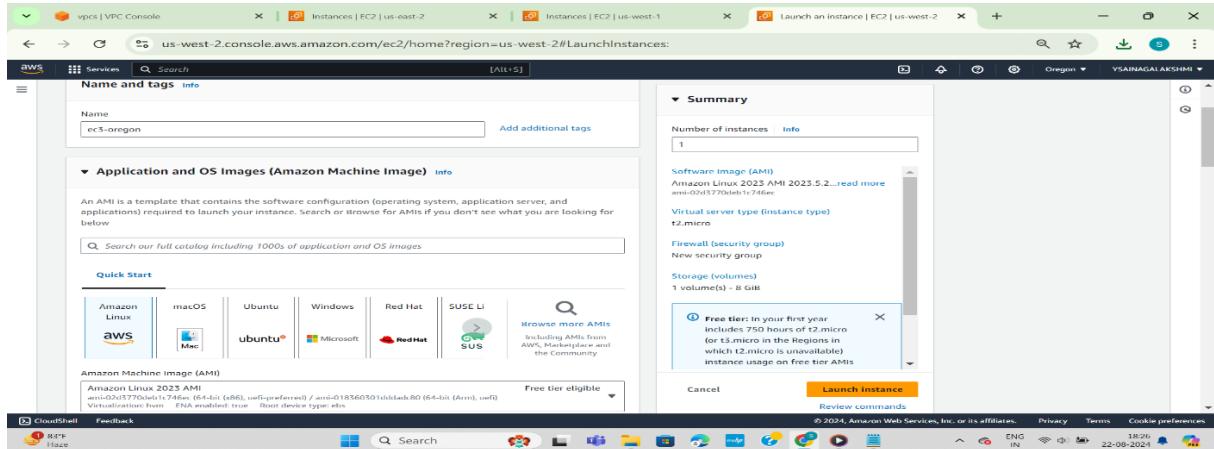
The screenshot shows the AWS VPC Console interface. The main window displays the 'Instances (1) Info' section. One instance is listed: 'ec2-ohio' (Instance ID: i-00002fd1e12c648f2), which is currently 'Running'. The instance type is 't2.micro' and its status is 'Initializing'. Other columns include 'Name', 'Instance ID', 'Instance state', 'Status check', 'Alarm status', 'Availability Zone', and 'Public IPv4 DNS'. A 'Launch instances' button is visible at the top right. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, and CloudShell.

CALIFORNIA:

This screenshot shows the 'Launch Instances' wizard in the AWS VPC Console for the us-west-1 region. The current step is 'Name and Tags'. The instance is named 'ec2-california'. The summary on the right indicates 1 instance will be launched using the 'Amazon Linux 2023 AMI (ami-04feade22817d69)' AMI, with a 't2.micro' instance type. It also shows a free tier warning about 750 hours of t2.micro usage. Other summary details include a new security group, 1 volume (2 GiB), and a launch command. Buttons for 'Cancel' and 'Launch instance' are present at the bottom.

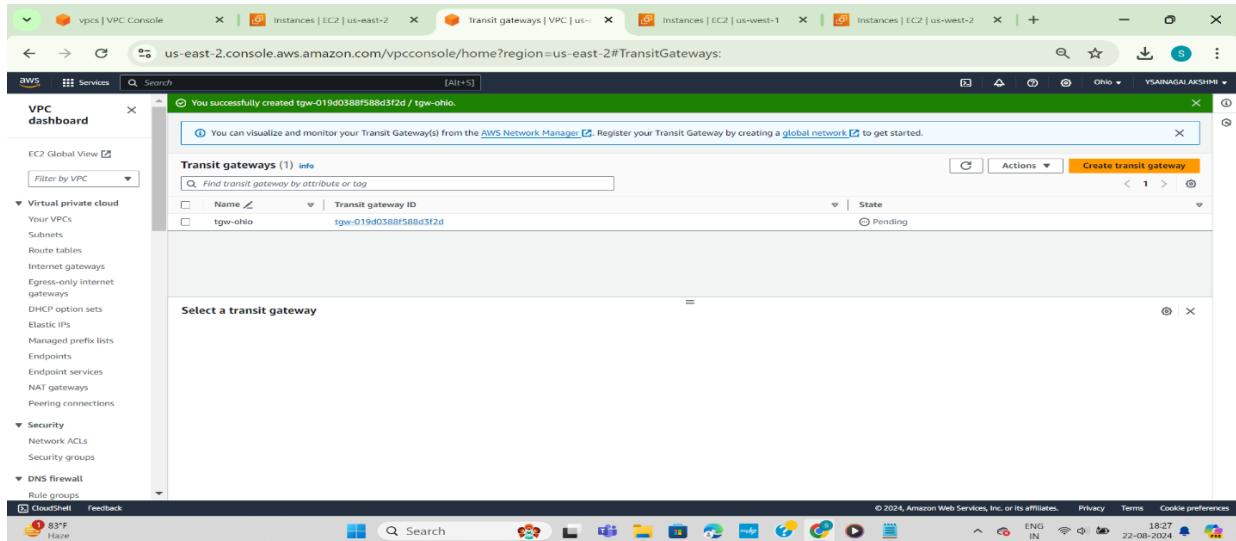
The screenshot shows the AWS VPC Console interface for the us-west-1 region. The main window displays the 'Instances (1) Info' section. One instance is listed: 'ec2-california' (Instance ID: i-0f71rf47954f5e5a), which is currently 'Pending'. The instance type is 't2.micro'. Other columns include 'Name', 'Instance ID', 'Instance state', 'Status check', 'Alarm status', 'Availability Zone', and 'Public IPv4 DNS'. A 'Launch instances' button is visible at the top right. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, and CloudShell. The status bar at the bottom shows '83°F Haze'.

OREGON:



5.Create Transit Gateway for all three regions.

OHIO:



CALIFORNIA:

You successfully created tgw-0267bes0ad7890059 / tgw-california.

Transit gateways (1) info

Name	Transit gateway ID	State
tgw-california	tgw-0267bes0ad7890059	Pending

Select a transit gateway

OREGON:

You successfully created tgw-019a2t4d5t090665 / tgw-oregon.

Transit gateways (1) info

Name	Transit gateway ID	State
tgw-oregon	tgw-019a2t4d5t090665	Pending

Select a transit gateway

6. Create 2 transit gateway attachments for each region.

- One for vpc
 - Other one for peering connection
- VPC and Peering Connections between the regions:
- Ohio – California
 - California – Oregon
 - Oregon – Ohio

VPC Connections:

You successfully created tgw-attach-03754ffcc579cc03ba / tgw-attach-03754ffcc579cc03ba.

Transit gateway attachments (1) info

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID	Association route table ID
tgwa-ohio-california	tgw-attach-03754ffcc579cc03ba	tgw-019a2t4d5t090665	Pending	VPC	vpc-062cf573e632027400	-

Select a transit gateway attachment

Screenshot of the AWS VPC console showing the creation of a Transit Gateway Attachment between a VPC and a Transit Gateway.

Transit gateway attachments (1) info

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID
tgw-california-oregon	tgw-attach-045feccaf763081fb	tgw-026/bec0ad/7e50059	Pending	VPC	vpc-021602f0/502d3d8c

Select a transit gateway attachment

Screenshot of the AWS VPC console showing the creation of a Transit Gateway Attachment between a VPC and a Transit Gateway.

Transit gateway attachments (1) info

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID
tgw-oregon-ohio	tgw-attach-0ffeb31b550b7655f5	tgw-01f9a374d3f09613	Pending	VPC	vpc-0f1a2dc7a221bd90f

Select a transit gateway attachment

Peering Connections:

Screenshot of the AWS VPC console showing the creation of a Peering Connection between two Transit Gateways.

Transit gateway attachments (1/2) info

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID	Association route table
tgw-ohio-california-peering	tgw-attach-05fcfe047a7afe80c	tgw-019a03bf80bf03f2d	Initiating Request	Peering	tgw-026/bec0ad/7e50059	
<input checked="" type="checkbox"/> tgw-ohio-california-vpc	<input checked="" type="checkbox"/> tgw-attach-03754ffe579ce03ba	<input checked="" type="checkbox"/> tgw-019a03bf80bf03f2d	Available	VPC	vpc-0b2a974d2d2740e	tgw-01b-01322d

Transit gateway attachment: tgw-attach-03754ffe579ce03ba / tgw-ohio-california

Details

Transit gateway attachment ID: tgw-attach-03754ffe579ce03ba	Transit gateway ID: tgw-019a03bf80bf03f2d	Transit gateway owner ID: 692859906022	Subnet IDs: subnet-022942d461656f9
State: Available	Resource owner ID: 692859906022	DNS support: Enable	Resource type: VPC
Resource ID: vpc-0b2a974d2d2740e	IPv6 support: Disable	Association state: Associated	Association route table ID: tgw-rtb-01322d81990566355
Appliance Mode support: Disabled			

Screenshot of the AWS VPC console showing the creation of a Peering Connection between two Transit Gateways.

Transit gateway attachments (1/3) info

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID
<input checked="" type="checkbox"/> tgwa-california-oregon-peering	<input checked="" type="checkbox"/> tgw-attach-05fcfe047a7afe80c	<input checked="" type="checkbox"/> tgw-026/bec0ad/7e50059	Pending Acceptance	Peering	tgw-019d038f588d3f2
<input checked="" type="checkbox"/> tgwa-california-oregon-vpc	<input checked="" type="checkbox"/> tgw-attach-045feccaf763081fb	<input checked="" type="checkbox"/> tgw-026/bec0ad/7e50059	Initiating Request	Peering	tgw-019a03bf80bf03f2d
		<input checked="" type="checkbox"/> tgw-026/bec0ad/7e50059	Available	VPC	vpc-021602f0/502d3d8c

Transit gateway attachment: tgw-attach-045feccaf763081fb / tgwa-california-oregon

Screenshot of the AWS VPC console showing the successful creation of a peering attachment between two Transit Gateways. The attachment is named tgw-attach-05a1dd016148219 and is associated with tgw-oregon-ohio.

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID
tgw-oregon-ohio-peering	tgw-attach-05a1dd016148219	tgw-01f9a2fd4d9f096b2	Pending Acceptance	Peering	tgw-019d0398f98d3f2d
tgw-attach-08ec51b550b7655f5	tgw-attach-08ec51b550b7655f5	tgw-01f9a5f4d51096b5	Pending Acceptance	Peering	tgw-0267bdca0d780005
tgw-oregon-ohio-vpc	tgw-attach-08ec51b550b7655f5	tgw-01f9a5f4d51096b5	Available	VPC	vpc-0f1c7de7a22bbba90f

Accept Transit Gateway Attachments:

Screenshot of the AWS VPC console showing the list of Transit gateway attachments. The attachment tgw-attach-05a1dd016148219 is selected for acceptance.

Name	Transit gateway attachment ID	Transit gateway ID	State	Resource type	Resource ID	Association route
tgw-oregon-california-peering	tgw-attach-05a1dd016148219	tgw-019d0398f98d3f2d	Pending	Peering	tgw-019d0398f98d3f2d	-
tgw-oregon-california-vpc	tgw-attach-0575dfe579cd35a	tgw-019d0398f98d3f2d	Available	VPC	vpc-0b2ef92a6d22d20e	targett0152-22

Screenshot of the AWS VPC console showing the details of the transit gateway attachment tgw-attach-05fc6c847a9fe880c. The 'Actions' menu is open, showing the option to 'Accept transit gateway attachment'.

Screenshot of the AWS VPC console showing the details of the transit gateway attachment tgw-attach-0dc87fa8030451e2b. The 'Actions' menu is open, showing the option to 'Accept transit gateway attachment'.

- At all regions status should be in available state.

Name	Transit gateway attachment ID	Transit gateway ID	Status	Resource type	Resource ID
tgw-oregon-ohio-peering	tgw-attach-05a1dbf116148219	tgw-01f0a3fd6153096ba3	Available	Peering	tgw-019b0588f584512d
tgw-attach-05a1dbf116148219	tgw-attach-05a1dbf116148219	tgw-01f0a3fd6153096ba3	Available	Peering	tgw-026fbec0af29930259
tgw-oregon-ohio-vpc	tgw-attach-05a1dbf116148219	tgw-01f0a3fd6153096ba3	Available	VPC	vpc-0f3a7de2a22bba90f

7. Go to EC2 instance and connect the server git bash in all three regions.

- sudo -i
- yum update -y && yum install nginx -y && cd /usr/share/nginx/html
- ls
- rm index.html
- yes
- vi index.html
 - ✓ press i to insert the data.
- systemctl restart nginx
- copy the public ip
- curl public ip :80 ---- the data which was inserted will be shown here.
VPCs will be connected through the transit gateway.
- systemctl status nginx

OHIO – CALIFORNIA:

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-4-197 ~]$ sudo -i
[root@ip-10-0-4-197 ~]# yum update -y
last metadata expiration check: 0:26:06 ago on Thu Aug 22 12:53:21 2024.
Dependencies resolved.
Nothing to do.
Preparing transaction metadata...
[root@ip-10-0-4-197 ~]# yum install nginx -y
last metadata expiration check: 0:26:16 ago on Thu Aug 22 12:53:21 2024.
Dependencies resolved.

Transaction Summary
install 7 Packages

I-00023d1e122cb48f2 (ec1-ohio)
PublicIPs: 18.221.128.258 PrivateIPs: 10.0.4.197

CloudShell Feedback
81% Haze

```

CALIFORNIA - OREGON:

```
aws Services Search [Alt+5] N. California VSAINAGALAKSHMI

Amazon Linux 2023
http://aws.amazon.com/linux/amazon-linux-2023

/mf
[ec2-user@ip-10-50-13-76 ~]$ sudo yum update -y & yum install nginx -y
Last metadata expiration check: 0:30:57 ago on Thu Aug 22 12:55:29 2024.
Dependencies resolved.
Nothing to do.
Nothing to do.
Last metadata expiration check: 0:30:58 ago on Thu Aug 22 12:55:29 2024.
Dependencies resolved.

Package           Architecture      Version          Repository
installing:
nginx              x86_64          1:1.24.0-1.amzn2023.0.2  amazonlinux
Installing dependencies:
  generic-logs-nginx
    noarch          18.0.0-12.amzn2023.0.3  amazonlinux
  gperftools-libs
    x86_64          2.9.1-1.amzn2023.0.3  amazonlinux
  libcurl
    x86_64          1.1.1-1.amzn2023.0.2  amazonlinux
  nginx-core
    x86_64          1:1.24.0-1.amzn2023.0.2  amazonlinux
  nginx-fsleystem
    noarch          1:1.24.0-1.amzn2023.0.2  amazonlinux
  nginx-mimetypes
    noarch          2.1.9-3.amzn2023.0.3  amazonlinux

i-0F717f479534f3e3a (ec2-california)
PublicIPs: 15.57.40.215 PrivateIPs: 10.50.13.76
```

Screenshot of AWS CloudShell showing the configuration and deployment of Nginx on an Amazon Linux 2023 instance.

```

Installing : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
Installing : libcurl-1:1.24.0-1.amzn2023.0.2.x86_64
Running auripihuk: nginx-1:1.24.0-1.amzn2023.0.2.x86_64
Verifying  : generic-logos-httpsd-18.0.0-12.amzn2023.0.3.noarch
Verifying  : openrptools-libs-2.9.1-1.amzn2023.0.3.x86_64
Verifying  : libunwind-1:1.4.0-5.amzn2023.0.2.x86_64
Verifying  : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
Verifying  : nginx-filcystem-1:1.24.0-1.amzn2023.0.2.noarch
Verifying  : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Installed:
generic-logos-httpsd-18.0.0-12.amzn2023.0.3.noarch
nginx-1:1.24.0-1.amzn2023.0.2.x86_64
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-10-50-13-76 ~]# cd /usr/share/nginx/html
[root@ip-10-50-13-76 html]\# ls
404.html 50x.html  index.html  nginx-logo.png  poweredby.png
[root@ip-10-50-13-76 html]\# curl 13.57.40.2.215:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome</title>
</head>
<body>
<h1>Welcome</h1>
</body>
</html>
[root@ip-10-50-13-76 html]\# vi index.html
[root@ip-10-50-13-76 html]\# systemctl restart nginx
[root@ip-10-50-13-76 html]\# curl 13.57.40.2.215:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome</title>
</head>
<body>
<h1>Welcome</h1>
</body>
</html>
[root@ip-10-50-13-76 html]\# curl 13.57.40.2.215:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome</title>
</head>
<body>
<h1>Welcome</h1>
</body>
</html>
[root@ip-10-50-13-76 html]\# curl 13.57.40.2.215:80
This is connection between the California - Ohio
[root@ip-10-50-13-76 html]\#

```

i-0f71f479534f3e3a (ec2-2-california)
PublicIPs: 15.57.40.215 PrivateIPs: 10.50.13.76

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 18:59 22-09-2024

Screenshot of AWS CloudShell showing the configuration and deployment of Nginx on an Amazon Linux 2023 instance.

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Thu Aug 22 13:26:05 2024 from 13.52.6.116
[ec2-user@ip-10-50-13-76 ~]# 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 Thu 2024-08-22 13:20:31 UTC; 32min ago
     Process: 26076 ExecStartPre=/usr/bin/mk_f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 26077 ExecStart=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 26092 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
Main PID: 26912 (nginx)
   Tasks: 2 (limit: 1112)
      Memory: 2M
        CPU: 50ms
       CGroup: /system.slice/nginx.service
           ├─26912 "nginx: master process /usr/sbin/nginx"
           └─26913 "nginx: worker process"

Aug 22 13:28:31 ip-10-50-13-76.us-west-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 22 13:28:31 ip-10-50-13-76.us-west-1.compute.internal nginx[26884]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 22 13:28:31 ip-10-50-13-76.us-west-1.compute.internal nginx[26884]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 22 13:28:31 ip-10-50-13-76.us-west-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-50-13-76 ~]#

```

i-0f71f479534f3e3a (ec2-2-california)
PublicIPs: 15.57.40.215 PrivateIPs: 10.50.13.76

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 19:21 22-08-2024

OREGON - OHIO:

Screenshot of AWS CloudShell showing the configuration and deployment of Nginx on an Amazon Linux 2023 instance.

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-60-3-174 ~]# sudo -i
[ec2-user@ip-10-60-3-174 ~]# yum update -y && yum install nginx -y && cd /usr/share/nginx/html
[ec2-user@ip-10-60-3-174 ~]# metadata expiration check: 0:32:39 ago on Thu Aug 22 12:58:18 2024.
Dependencies resolved.
Nothing to do.
Complete!
last metadata expiration check: 0:32:39 ago on Thu Aug 22 12:58:18 2024.
Dependencies resolved.

  Package          Architecture Version           Repository
Installling:
nginx            x86_64      1:1.24.0-1.amzn2023.0.2   amazonlinux
Installling Dependencies:
generic-logoshttpsd      noarch    10.0.0.12.amzn2023.0.3   amazonlinux
openrptools-libs      x86_64      2.9.1.1.amzn2023.0.3   amazonlinux
libunwind            x86_64      1.4.0.5.amzn2023.0.2   amazonlinux
nginx-core           x86_64      1:1.24.0-1.amzn2023.0.2   amazonlinux
nginx-filcystem       noarch    2.1.49-3.amzn2023.0.2   amazonlinux
nginx-mimetypes       noarch    2.1.49-3.amzn2023.0.3   amazonlinux

i-006c1ccc95470ebdd (ec2-oregon)
PublicIPs: 54.202.27.16 PrivateIPs: 10.60.3.174

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 19:02 22-08-2024

```

```

aws Services Search [Alt+S] Oregon YSAINAGALAKSHMI
installing : nginx-filccsystem-1:1.24.0-1.amzn2023.0.2.noarch
installing : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
installing : libunwind-1.4.0-5.amzn2023.0.2.x86_64
installing : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
installing : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
installing : generic-openssl-https-10.0.0.12.amzn2023.0.1.noarch
installing : nginx-1:1.24.0-1.amzn2023.0.2.x86_64
Running scriptlets: nginx-1:1.24.0-1.amzn2023.0.2.x86_64
Verifying : generic-openssl-https-10.0.0-12.amzn2023.0.3.noarch
Verifying : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Verifying : libunwind-1.4.0-5.amzn2023.0.2.x86_64
Verifying : nginx-1:1.24.0-1.amzn2023.0.2.x86_64
Verifying : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
Verifying : nginx-filccsystem-1:1.24.0-1.amzn2023.0.2.noarch
Verifying : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Installed:
generic-openssl-https-10.0.0.12.amzn2023.0.1.noarch
nginx-1:1.24.0-1.amzn2023.0.2.x86_64
nginx-filccsystem-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-10-60-3-174 ~]# ls
404.html 50x.html icoons index.html nginx-logo.png poweredby.png
[root@ip-10-60-3-174 ~]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-10-60-3-174 ~]# vi index.html
[root@ip-10-60-3-174 ~]# systemctl restart nginx
[root@ip-10-60-3-174 ~]# curl 54.202.27.16:80
This is a connection between origin -> id
[root@ip-10-60-3-174 ~]#

```

i-006e1ccc95470ebdd (ec3-oregon)
PublicIPs: 54.202.27.16 PrivateIPs: 10.60.3.174

```

aws Services Search [Alt+S] Oregon YSAINAGALAKSHMI
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Thu Aug 22 13:56:38 UTC 2024 from 18.237.140.165
[ec2-user@ip-10-60-3-174 ~]# 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 Thu 2024-08-22 13:31:56 UTC; 30min ago
     Process: 27332 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 27333 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 27341 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 27343 (nginx)
      Tasks: 1 (limit: 1112)
     Memory: 2.2M
        CPU: 59ms
       CGroup: /system.slice/nginx.service
           └─27363 "nginx: master process /usr/sbin/nginx"
              ├─27364 "nginx: worker process"

Aug 22 13:31:56 ip-10-60-3-174.us-west-2.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 22 13:31:56 ip-10-60-3-174.us-west-2.compute.internal nginx[27333]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 22 13:31:56 ip-10-60-3-174.us-west-2.compute.internal nginx[27333]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 22 13:31:56 ip-10-60-3-174.us-west-2.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-60-3-174 ~]#

```

i-006e1ccc95470ebdd (ec3-oregon)
PublicIPs: 54.202.27.16 PrivateIPs: 10.60.3.174