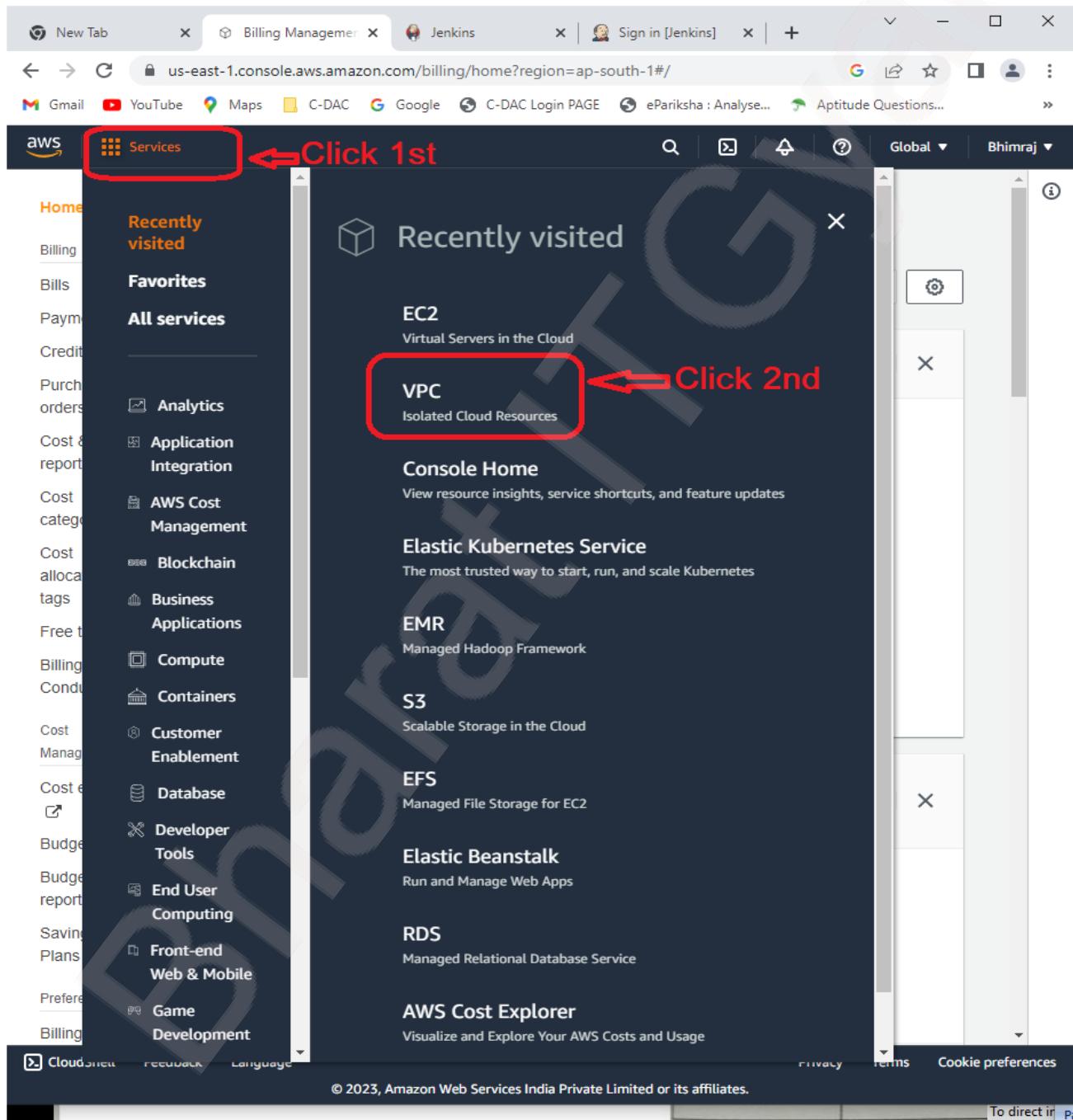


Scenario:

Create VPC (10.0.0.0/16) create two subnets 1 is 10.0.1.0/24 and subnet2 is 10.0.5.0/24 attach these subnets to 2 Ubuntu Machines and try to ping each other.

In the Services Dashboard, click on "VPC."



In the VPC Dashboard, click on "Create VPC."

The screenshot shows the AWS VPC Management Dashboard. On the left, there's a sidebar with various navigation options like 'Virtual private cloud', 'Security', 'DNS firewall', and 'Network Firewall'. The main area displays 'Resources by Region' for the Asia Pacific region. A prominent orange button labeled 'Create VPC' is highlighted with a red box and an arrow pointing to it. Below the button, there's a 'Launch EC2 Instances' button and a note stating 'Note: Your Instances will launch in the Asia Pacific region.' At the bottom, there are links for 'CloudShell', 'Feedback', 'Language', 'Privacy', 'Terms', and 'Cookie preferences'.

Select "VPC only" and proceed to create the VPC.

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 - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC-first

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

10.0.0.0/16

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy [Info](#)
Default

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key **Name** Value - optional **VPC-first** Remove

Add new tag

You can add 49 more tags.

Create VPC

The VPC has been created successfully.

The screenshot shows the AWS VPC Management console. A success message 'You successfully created vpc-0d175a744fd3daaaa / VPC-first' is displayed at the top. The main title 'vpc-0d175a744fd3daaaa / VPC-first' is also highlighted with a red box. The 'Details' section provides the following information:

VPC ID	State	DNS hostnames	DNS resolution
vpc-0d175a744fd3daaaa	Available	Disabled	Enabled
Tenancy	DHCP option set	Main route table	Main network ACL
Default	dopt-0e986bed27945b2b2	rtb-047b8563a8a45ee70	acl-065044fdd8f6acdf3
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network border group)
No	10.0.0.0/16	—	—
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID	—
Disabled	—	111698785569	—

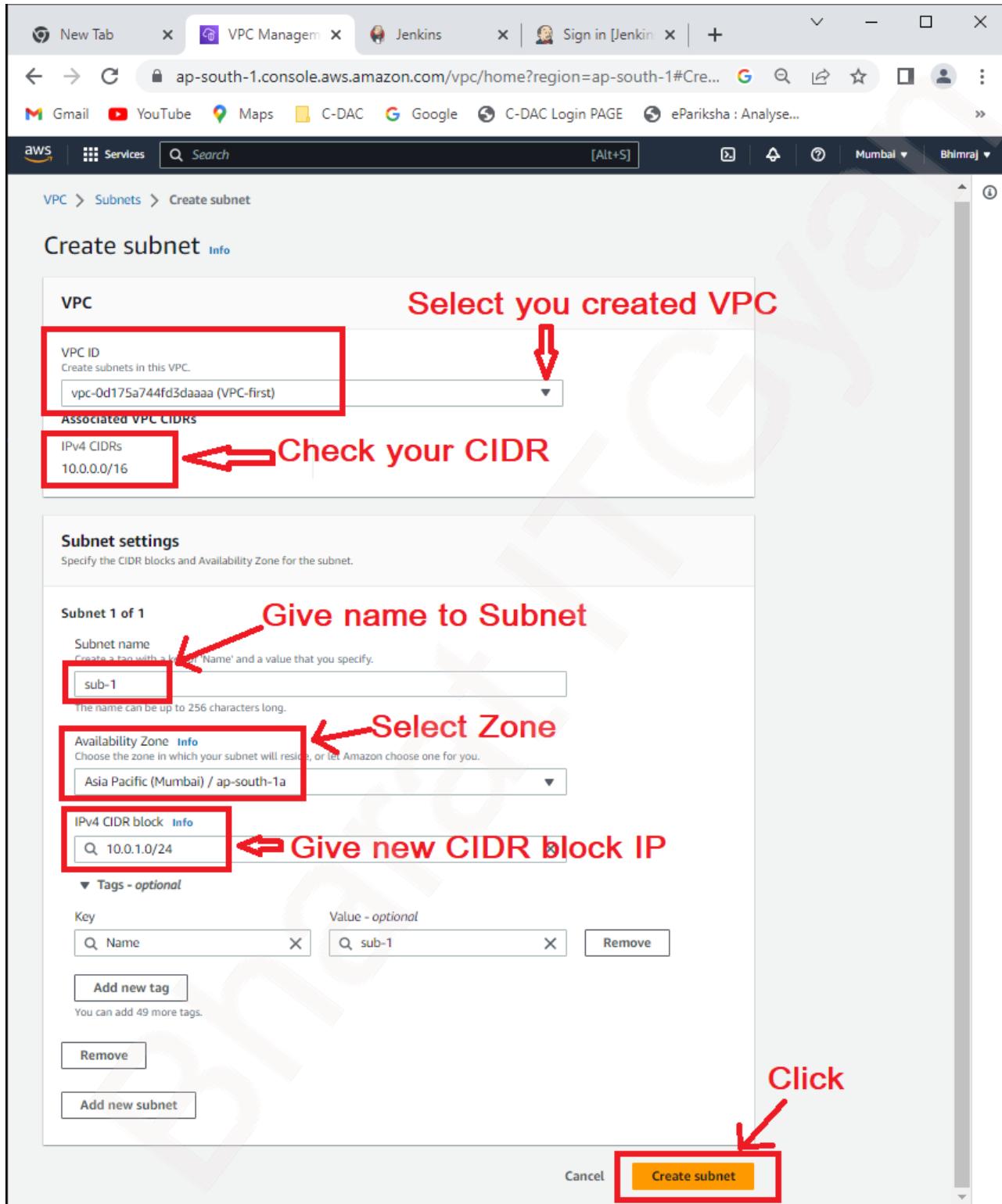
The 'Resource map' section shows three components: 'VPC', 'Subnets (0)', and 'Route tables (1)'. A tooltip box titled 'Introducing the VPC resource map' explains the features of the resource map.

We need to create two distinct subnet networks.

The screenshot shows the AWS VPC Subnets page. On the left, a sidebar menu is open under the 'Subnets' section, with the 'Subnets' item highlighted by a red box and labeled '1st'. At the top right, there is a 'Create subnet' button highlighted by a red box and labeled '2nd'.

Name	Subnet ID	State	VPC	IPv4 CIDR Range
-	subnet-0e847e21d3f7d90b4	Available	vpc-0e0b9bbe071cea9d9	172.31.12.0/24
-	subnet-05770eab6b511f78b	Available	vpc-0e0b9bbe071cea9d9	172.31.13.0/24
-	subnet-0726258c99db5f537	Available	vpc-0e0b9bbe071cea9d9	172.31.14.0/24

Create the first subnet.



Sub-1 created successfully.

The screenshot shows the AWS VPC Subnets page. A green success message at the top states: "You have successfully created 1 subnet: subnet-0fb5f52b9aae54525". Below this, a table lists one subnet: "sub-1" with Subnet ID "subnet-0fb5f52b9aae54525", State "Available", and VPC "vpc-0d175a744fd3daaaa". Red arrows point from the success message and the table row to the text "Successfully created sub-1" overlaid on the page.

Subnets (1) Info

Subnet ID: subnet-0fb5f52b9aae54525 X Clear filters

Name	Subnet ID	State	VPC
sub-1	subnet-0fb5f52b9aae54525	Available	vpc-0d175a744fd3daaaa

Successfully created sub-1

Subnets (1) Info

Subnet ID: subnet-0fb5f52b9aae54525 X Clear filters

Name	Subnet ID	State	VPC
sub-1	subnet-0fb5f52b9aae54525	Available	vpc-0d175a744fd3daaaa

Successfully created sub-1

Create second subnet.

The screenshot shows the 'Create Second Subnet' step in the AWS VPC Manager. The interface is divided into two main sections: 'VPC' and 'Subnet settings'.

VPC Section:

- VPC ID:** A dropdown menu showing 'vpc-0d175a744fd3daaaa (VPC-first)'. A red box highlights this field, and a red arrow points to it with the text 'Check your VPC'.
- IPv4 CIDRs:** A dropdown menu showing '10.0.0.0/16'. A red box highlights this field, and a red arrow points to it with the text 'Check CIDR'.

Subnet settings Section:

- Subnet 1 of 1:**
 - Subnet name:** An input field containing 'sub-2'. A red box highlights this field, and a red arrow points to it with the text 'Give name to Second Subnet'.
 - Availability Zone:** A dropdown menu showing 'Asia Pacific (Mumbai) / ap-south-1b'. A red box highlights this field, and a red arrow points to it with the text 'Select another Zone'.
 - IPv4 CIDR Block:** An input field showing 'Q 10.0.2.0/24'. A red box highlights this field, and a red arrow points to it with the text 'Give different CIDR Block'.
- Tags - optional:** A section where a tag 'Name: sub-2' is added. A red box highlights this entire section, and a red arrow points to it with the text 'Click here'.
- Buttons:** At the bottom right of the form area, there are 'Cancel' and 'Create subnet' buttons. The 'Create subnet' button is highlighted with a red box and a red arrow pointing to it with the text 'Click here'.

Sub-2 created successfully.

The screenshot shows the AWS VPC Management console. In the top right corner, there is a green success message: "You have successfully created 1 subnet: subnet-0f146c7ced91d027c". Below this message, the "Subnets (5)" table is displayed. The row for "sub-2" is highlighted with a red box and has a red arrow pointing from it to a large black-outlined box containing the text "Successfully created sub-2". The "sub-2" row contains the following information:

Name	Subnet ID	State	VPC
sub-2	subnet-0f146c7ced91d027c	Available	vpc-0d175a744fd3daaaa
sub-1	subnet-0fb5f52b9aae54525	Available	vpc-0d175a744fd3daaaa
-	subnet-0726258c99db5f537	Available	vpc-0e0b9bbe071cea9d9
-	subnet-0e847e21d3f7d90b4	Available	vpc-0e0b9bbe071cea9d9
-	subnet-05770eab6b511f78b	Available	vpc-0e0b9bbe071cea9d9

The left sidebar shows the navigation menu for VPC management, including sections for Virtual private cloud, Security, DNS firewall, Network Firewall, and Virtual private network (VPN). The "Subnets" section is currently selected.

We need to create internet gateway to access the internet and connect the Ubuntu machines publicly. Click “Create Internet Gateway”

The screenshot shows the AWS VPC Internet Gateways page. On the left, there is a navigation sidebar with various VPC-related options like VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud, Internet gateways (which is selected and highlighted with a red box), Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. Below these are sections for Security (Network ACLs, Security groups), DNS Firewall (Rule groups, Domain lists), Network Firewall (Firewalls, Firewall policies, Network Firewall rule groups, TLS inspection configurations), and Virtual private network (VPN) (Customer gateways, Virtual private). The main content area shows a table titled "Internet gateways (1/1)" with one item listed: "igw-09446d448c5eb21a1" (Name), "Attached" (State), and "vpc-0e0b9bbe071" (VPC ID). At the top right of the table is an "Actions" button with a red box containing the number "2" and a "Create internet gateway" button, also highlighted with a red box. A large red box labeled "1" is placed over the "Internet gateways" link in the sidebar. A red box labeled "2" is placed over the "Create internet gateway" button in the Actions menu.

Give name to and click create Internet Gateway.

The screenshot shows the 'Create internet gateway' page in the AWS VPC console. The 'Name tag' field contains 'New-Internet-gateway'. A red box highlights this field with the instruction 'Give name to internet gateway'. Below it, the 'Tags - optional' section shows a single tag 'Name: New-Internet-gateway'. A red box highlights this section with the instruction 'click here'. At the bottom right is a large red box highlighting the 'Create internet gateway' button.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

New-Internet-gateway

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - optional

Name New-Internet-gateway Remove

Add new tag

You can add 49 more tags.

Create internet gateway

Internet Gateway created.

The screenshot shows the AWS VPC Management console with a success message: "The following internet gateway was created: igw-05a6351d900a3c038 - New-Internet-gateway. You can now attach to a VPC to enable the VPC to communicate with the internet." A red box highlights this message. To the right, another red box highlights the "Attach to a VPC" button. A large red arrow points from the text "Next Attach you created VPC to here" to the "Attach to a VPC" button. The central area displays the Internet gateway details: ID igw-05a6351d900a3c038, State Detached, VPC ID -, Owner 111698785569, and Tag Name New-Internet-gateway. The left sidebar lists various VPC-related services like EC2 Global View, Virtual private cloud, Security, DNS firewall, Network Firewall, and Virtual private network (VPN). The bottom navigation bar includes CloudShell, Feedback, Language, and links to Privacy, Terms, and Cookie preferences.

The following internet gateway was created: igw-05a6351d900a3c038 - New-Internet-gateway. You can now attach to a VPC to enable the VPC to communicate with the internet.

Attach to a VPC

Next Attach you created VPC to here

Internet gateway Successfully created

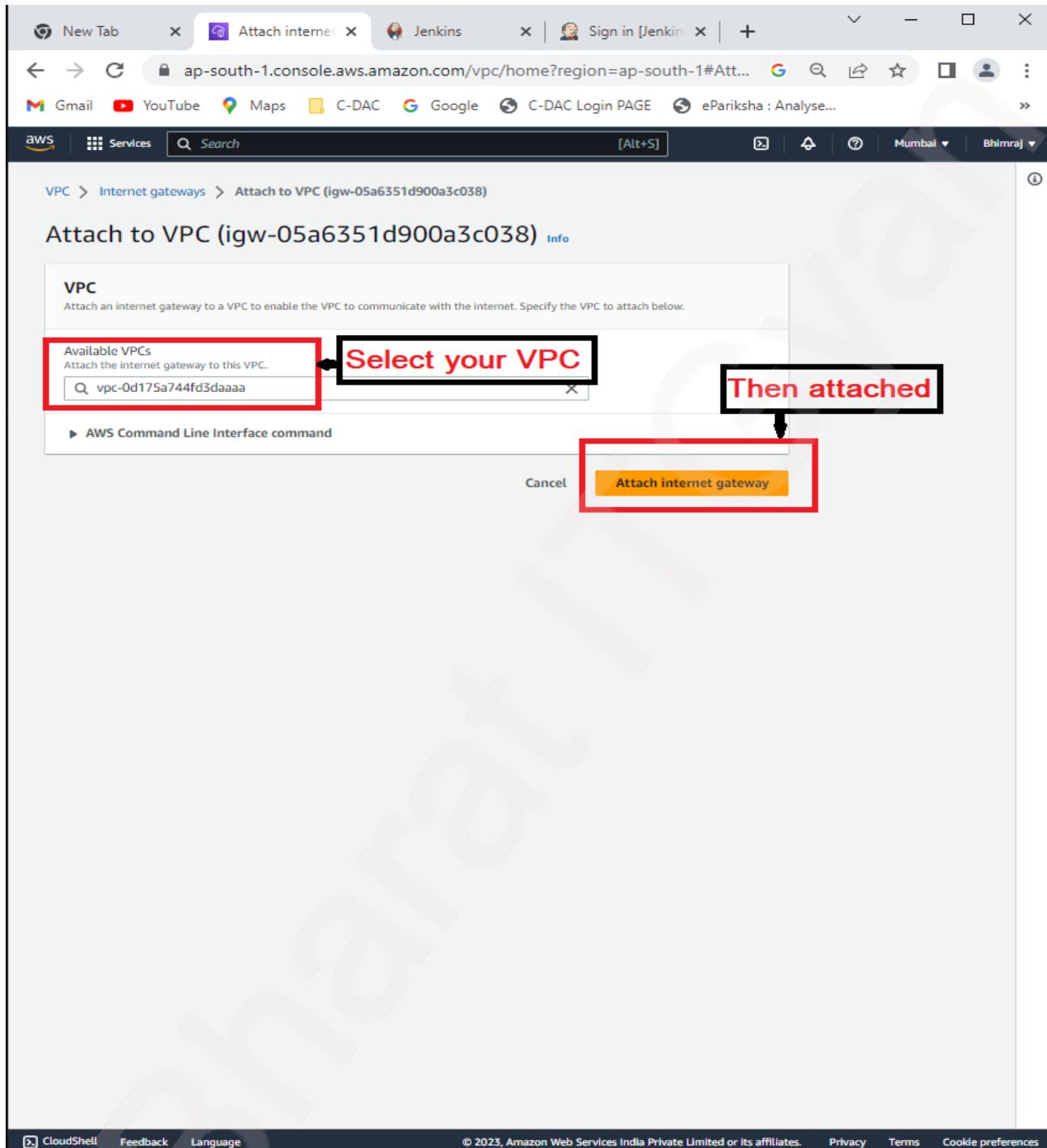
Internet gateway ID	State	VPC ID	Owner
igw-05a6351d900a3c038	Detached	-	111698785569

Tags

Key	Value
Name	New-Internet-gateway

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

The Internet gateway is currently in a detached state. We need to attach it to the VPC.



The Internet gateway has been successfully attached to the VPC.

The screenshot shows the AWS VPC Management console with the URL ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#Internet-gateways. The left sidebar shows navigation options like VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud, Security, DNS firewall, Network Firewall, and Virtual private network (VPN). The main content area displays an Internet gateway named "igw-05a6351d900a3c038 / New-Internet-gateway". A green success message at the top states: "Internet gateway igw-05a6351d900a3c038 successfully attached to vpc-0d175a744fd3daaaa". Below this, a red box highlights the "Successfully attached to VPC" status. Another red box highlights the "Attached" state under the "Details" section, which also lists the Internet gateway ID (igw-05a6351d900a3c038), VPC ID (vpc-0d175a744fd3daaaa), and Owner (111698785569). The "Tags" section shows a single tag named "Name" with the value "New-Internet-gateway".

To enable private IPv4 traffic between EC2 instances in peered VPCs, we need to create route tables for the peering VPC connection.

The screenshot shows the AWS VPC Route Tables page. On the left, a sidebar lists various VPC-related services. The 'Route tables' link is highlighted with a red box and labeled '1'. In the main content area, the 'Route tables (2)' section is displayed. A red box highlights the 'Actions' button, which has a red arrow pointing to it from the sidebar. Another red box highlights the 'Create route table' button. A large red box labeled 'Create new Route' is overlaid on the bottom right of the main content area. The table in the center lists two route tables with their respective IDs: rtb-01bdcaa42b30112ee and rtb-047b8563a8a45ee70.

Name	Route table ID	Explicit subnet associations	Edge associations
-	rtb-01bdcaa42b30112ee	-	-
-	rtb-047b8563a8a45ee70	-	-

Create a route table and assign a name to it. Then, select your VPC for the route table.

Give name to route table

Select your VPC

Click here

Create route table

The route table has been created successfully.

A screenshot of the AWS VPC Management console. The URL in the browser is `ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#Ro...`. The page shows a success message: "Route table rtb-0d02a4262454e9f57 | my-vpc-route was created successfully." A red arrow points from this message to the main content area. The main content area displays the details of the newly created route table, including its ID (rtb-0d02a4262454e9f57), VPC (vpc-0d175a744fd3daaaa), and owner ID (111698785569). The status is listed as "Active". Below this, a table titled "Routes (1)" shows one route entry: Destination 10.0.0.0/16, Target local, Status Active, and Propagated No. The left sidebar contains a navigation menu with categories like VPC dashboard, EC2 Global View, Virtual private cloud, Security, DNS firewall, Network Firewall, and Virtual private network (VPN).

Route table rtb-0d02a4262454e9f57 | my-vpc-route was created successfully.

VPC > Route tables > rtb-0d02a4262454e9f57 / my-vpc-route

my-vpc-route was created successfully

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0d02a4262454e9f57	No	-	-
VPC	Owner ID		
vpc-0d175a744fd3daaaa VPC-first	111698785569		

Routes Subnet associations Edge associations Route propagation Tags

Routes (1)					Edit routes
Filter routes		Both		< 1 >	
Destination	Target	Status	Propagated		
10.0.0.0/16	local	Active	No		

Please check your routing table to verify if the created subnet is present or not.

The screenshot shows the AWS VPC Manager interface. The left sidebar is collapsed, and the main area displays a route table named 'rtb-0d02a4262454e9f57 / my-vpc-route'. The 'Subnet associations' tab is selected. Two subnets are listed under 'Explicit subnet associations': 'sub-2' (subnet-0f146c7ced91...) and 'sub-1' (subnet-0fb5f52b9aae...). Both subnets have an IPv4 CIDR of 10.0.2.0/24 and 10.0.1.0/24 respectively. A red box highlights the list of subnets, and a red arrow points to a callout box at the bottom.

Check Your Subnet present or not in this route table

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
sub-2	subnet-0f146c7ced91...	10.0.2.0/24	-
sub-1	subnet-0fb5f52b9aae...	10.0.1.0/24	-

We need to create two Ubuntu machines in the EC2 instance to enable pinging between two different subnets in two networks.

The screenshot shows a browser window with multiple tabs open, including 'VPC Man', 'EC2 Glob', 'Jenkins', and 'Sign in'. The main content is the AWS Cloud Lab Practical interface. A red box labeled '1' highlights the 'Services' button in the top navigation bar. Another red box labeled '2' highlights the 'EC2' service card in the 'Recently visited' section. The left sidebar lists various services under the 'EC2' heading, such as Analytics, Application Integration, AWS Cost Management, Blockchain, Business Applications, Compute, Containers, Customer Enablement, Database, Developer Tools, End User Computing, Front-end Web & Mobile, Game Development, and Internet of Things. The right side of the screen displays a 'Recently visited' section with cards for VPC, EC2 (highlighted), Console Home, Elastic Kubernetes Service, EMR, S3, EFS, Elastic Beanstalk, RDS, and AWS Cost Explorer.

Go to EC2 and click on "Launch Instance."

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'Instances' section, the 'Launch instance' button is highlighted with a red box. The main content area displays various EC2 resources and their counts, including Instances (running) at 0, Auto Scaling Groups at 0, Dedicated Hosts at 0, Elastic IPs at 0, Instances at 0, Key pairs at 3, Load balancers at 0, Placement groups at 0, Security groups at 3, Snapshots at 0, and Volumes at 0. To the right, there are sections for 'Account attributes' (Supported platforms: VPC), 'Explore AWS' (Get Up to 40% Better Price Performance, Save up to 90% on EC2 with Spot Instances, Amazon GuardDuty Malware Protection), and 'Additional information' (Getting started guide, Documentation, All EC2 resources).

Create the first UbuntuPC1 machine.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The first step, 'Name and tags', has a red box around the 'Name' field containing 'UbuntuPC1'. A red bracket points from this field to a callout box labeled 'Give name to EC2 instance'. The second step, 'Application and OS Images (Amazon Machine Image)', has a red box around the 'Ubuntu' AMI card, which is highlighted with a blue border. A red bracket points from this card to a callout box labeled 'Select Ubuntu Image'. Below the AMI selection, there is descriptive text about the selected Ubuntu Server 22.04 LTS (HVM) AMI, including its ID and compatibility details. The bottom section of the wizard is partially visible, showing the 'Instance type' configuration.

Create a new key pair and download it.

The screenshot shows the AWS EC2 Launch Wizard interface. At the top, there are several tabs: New Tab, VPC Manager, EC2 Manage, Jenkins, Sign in [Jenk], and others like Gmail, YouTube, Maps, C-DAC, Google, C-DAC Login PAGE, ePariksha : Analyse..., Aptitude Questions..., and a sign-in for Mumbai and Bhimrao. Below the tabs, the main navigation bar has 'Services' selected. A search bar contains '[Alt+S]'. On the left, a sidebar lists sections: Key pair (login) Info, Network settings Info, Firewall (security groups) Info, and Configure storage Info. The 'Create new key pair' button is highlighted with a red box and an arrow pointing to it from below. The 'Network settings' section shows a dropdown menu with 'Select' and a 'Create new key pair' option. The 'Firewall (security groups)' section has a radio button selected for 'Create security group'. The 'Configure storage' section shows a root volume configuration with 1x 30 GiB gp2 selected. A note at the bottom says 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage'.

Name the key pair as "myvpc" and create the key pair.

The screenshot shows the AWS EC2 Management console with the 'Create key pair' dialog box open. The dialog box has a red box around the 'Key pair name' field, which contains 'myvpc'. A red arrow points from the text 'Give name to key-pair' to this field. Another red box surrounds the 'Create key pair' button at the bottom right of the dialog. A red arrow points from the text 'Click here' to this button. The background shows the 'Network settings' section of the instance configuration page, where a security group named 'myvpc' is being created.

Key pair name - required

Select Create new key pair

Network settings

Subnet: No preference (Default subnet in any)

Auto-assign public IP: Enable

Firewall (security groups): Create security group

We'll create a new security group called myvpc

Allow SSH traffic from

Allow HTTPS traffic from the internet

Allow HTTP traffic from the internet

Rules with source of 0.0.0.0/0 - security group rules to allow access

Create key pair

Key pair type: RSA

Private key file format: .pem

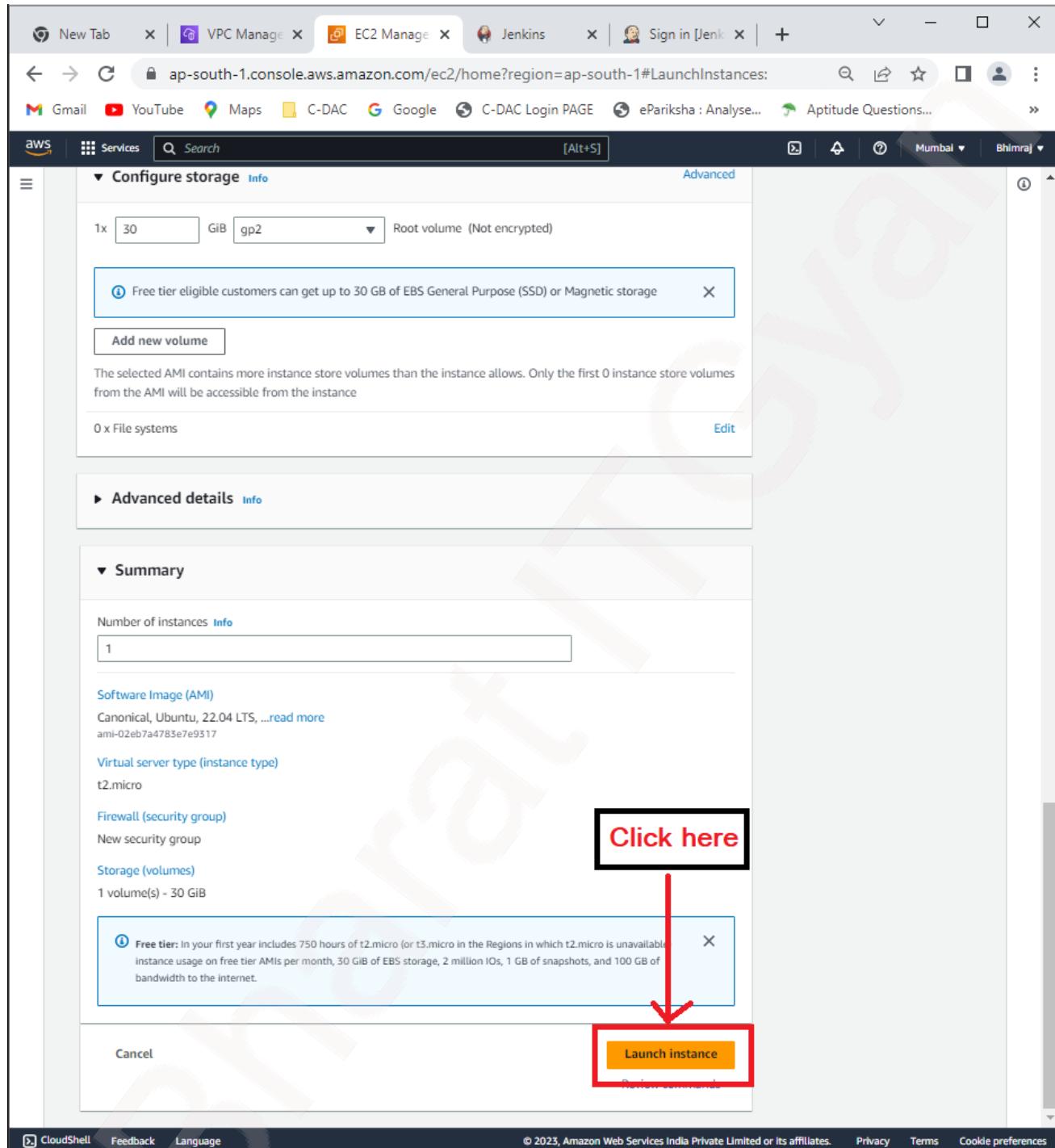
Configure storage: 1x 30 GB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

After creating the key pair, select the "myvpc" key. Then, check the option to allow HTTP traffic and set the storage to 30 GB.

The screenshot shows the AWS EC2 Launch Instance Wizard. The 'Key pair (login)' section has 'myvpc' selected in the dropdown, highlighted by a red box and labeled 'Select key pair'. The 'Network settings' section shows a selected VPC and subnet. In the 'Firewall (security groups)' section, the 'Create security group' option is selected, and a new security group named 'launch-wizard-2' is being configured. Under 'Allow HTTP traffic from the internet', the checkbox is checked and highlighted by a red box, with the text 'Allow this HTTP' overlaid. A warning message about allowing all IP addresses is visible. The 'Configure storage' section is highlighted by a red box and labeled 'Give 30 Gb Normal size', showing a 1x volume of 30 GiB using gp2 storage. The bottom navigation bar includes CloudShell, Feedback, Language, and links to Privacy, Terms, and Cookie preferences.

Then, click on "Launch Instance."



The EC2 instance has been created successfully.

The screenshot shows the AWS EC2 Instances Launch page. At the top, there is a success message: "Success Successfully initiated launch of instance (i-003436b424c687557)". Below this, a large red box highlights the message "EC2 Instance Created Successfully". The page also includes a "Next Steps" section with various options like "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", "Create EBS snapshot policy", "Manage detailed monitoring", "Create Load Balancer", "Create AWS budget", and "Manage CloudWatch alarms".

Success
Successfully initiated launch of instance (i-003436b424c687557)

EC2 Instance
Created Successfully

Next Steps

Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
[Create billing alerts](#)

Connect to your instance
Once your instance is running, log into it from your local computer.
[Connect to instance](#)
[Learn more](#)

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
[Connect an RDS database](#)
[Create a new RDS database](#)
[Learn more](#)

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots.
[Create EBS snapshot policy](#)

Manage detailed monitoring
Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.
[Manage detailed monitoring](#)

Create Load Balancer
Create a application, network gateway or classic Elastic Load Balancer.
[Create Load Balancer](#)

Create AWS budget
AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.
[Create AWS budget](#)

Manage CloudWatch alarms
Create or update Amazon CloudWatch alarms for the instance.
[Manage CloudWatch alarms](#)

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Create the second EC2 machine.

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar lists various services: New EC2 Experience, EC2 Dashboard, EC2 Global View, Events, Limits, 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 (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups), and Auto Scaling (Launch Configurations, Auto Scaling Groups). The main content area displays the 'Instances (1)' table. A single instance is listed: UbuntuPC1 (Instance ID: i-003436b424c687557). The instance is currently running. To the right of the table is a yellow button labeled 'Launch Instances'. A red box highlights the 'Launch Instances' button, and a red arrow points from the text 'Create 2nd EC2 Machine' to it. Another red box highlights the row for 'UbuntuPC1', and a red arrow points from the text '1st Machine' to it. At the bottom of the page, there is a 'Select an instance' dropdown menu. The footer contains links for CloudShell, Feedback, Language, Privacy, Terms, and Cookie preferences, along with a copyright notice for 2023, Amazon Web Services India Private Limited or its affiliates.

Name the EC2 instance as "UbuntuPC" and select the Ubuntu image.

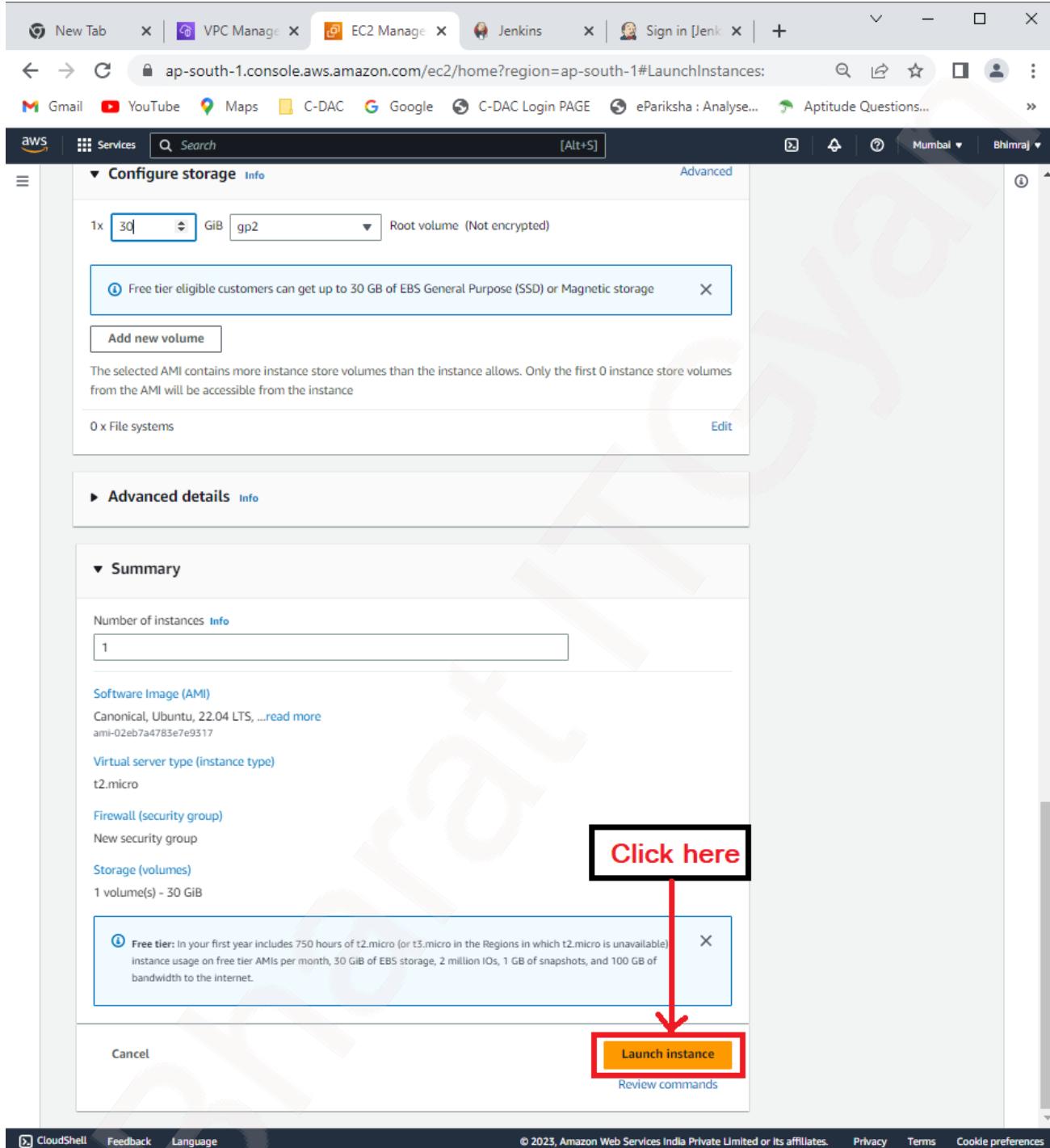
The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The first step, 'Name and tags', has a red box around the 'Name' field containing 'UbuntuPC2', with a red arrow pointing to it labeled 'Give name to EC2 instance'. The second step, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a grid of AMI icons. The 'Ubuntu' icon is highlighted with a red box and a red arrow pointing to it labeled 'Select ubuntu image'. Below the grid, details for 'Ubuntu Server 22.04 LTS (HVM), SSD Volume Type' are shown, including its AMI ID: ami-02eb7a4783e7e9317. The third step, 'Instance type', is partially visible at the bottom.

Select the existing key pair and check the box to allow HTTP traffic. Set the storage to 30 GB.

The screenshot shows the AWS EC2 Launch Wizard interface. The top navigation bar includes tabs for New Tab, VPC Manager, EC2 Manage, Jenkins, Sign in [Jenk], and others. The main content area is divided into sections:

- Key pair (login)**: A red box highlights the "Key pair name - required" input field containing "myvpc". A black box labeled "Select Old key-pair" is positioned above the "Create new key pair" button.
- Network settings**: Includes Network (Info), Subnet (Info), Auto-assign public IP (Info), and Firewall (security groups) (Info). Under Firewall, a red box highlights the "Allow HTTP traffic from the internet" checkbox, which is checked. A black box labeled "Allow HTTP" is placed next to it. A warning message below states: "⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." An arrow points from the "Allow HTTP" label to this warning message.
- Configure storage**: A red box highlights the "30" input field for volume size. A black box labeled "Give 30 Gb" is placed above the "1x 30 Gb gp2" configuration. A note below says: "💡 Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage".

Click on "Launch Instance."



The EC2 instance has been created successfully.

Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
UbuntuPC1	i-003436b424c687557	Running	t2.micro	2/2 checks passed	No alarms
UbuntuPC2	i-0be5c3462f192d51c	Running	t2.micro	Initializing	No alarms

2nd EC2 Instance created

Select an instance

New Tab VPC Manager Instances | EC2 Jenkins Sign in [Jenk

Gmail YouTube Maps C-DAC Google C-DAC Login PAGE ePariksha : Analyse... Aptitude Questions... Mumbai Bhlmraj

New EC2 Experience Tell us what you think

EC2 Dashboard EC2 Global View Events Limits

Instances Instances 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 Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

Load Balancing Load Balancers Target Groups

Auto Scaling Launch Configurations Auto Scaling Groups

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

We need to add an "All ICMP IPv4" rule to the security group of the EC2 instance. To do this, navigate to the EC2 instance's security group settings in the AWS Management Console and click on "Inbound Rules" or "Security Groups."

The screenshot shows the AWS Management Console interface for managing EC2 instances. The left sidebar contains various service links like EC2 Dashboard, Events, Limits, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the 'Instance summary' for an EC2 instance named 'i-0be5c3462f192d51c (UbuntuPC2)'. Key details shown include Instance ID, Public IP address (43.205.243.87), Instance state (Running), Hostname type, IP name, Answer private resource DNS name (IPv4 A), Auto-assigned IP address (43.205.243.87), IAM Role, VPC ID, Subnet ID, and Auto Scaling Group name. Below this, the 'Security' tab is selected (labeled '2'). Under 'Security details', the 'Security groups' section is highlighted (labeled '3') and shows a list containing 'sg-007362dba1958ced8 (launch-wizard-3)'. A callout box with the text 'Click on Storage groups' points to this entry.

Select "Security Groups" and edit the inbound rules for the storage group.

The screenshot shows the AWS EC2 Management console. The left sidebar is collapsed, and the main content area displays the details of a security group named 'sg-007362dba1958ced8 - launch-wizard-3'. The 'Inbound rules' tab is selected, showing two entries:

Name	Security group rule...	IP version	Type	Protocol
-	sgr-08807ba7966dcd9df	IPv4	HTTP	TCP
-	sgr-06acde44c220bacb6	IPv4	SSH	TCP

A red box highlights the 'Edit inbound rules' button at the top right of the table. A callout box with the text 'Click and Edit inbound rules' points to this button.

Add a rule by selecting "ALL ICMP-IPv4," then choose "Anywhere" under the IPv4 source, and save the rules.

Inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-08807ba7966dcd9df	HTTP	TCP	80	Custom ▾ <input type="text" value="0.0.0.0/0"/> X	<input type="button" value="Delete"/>
sgr-06acde44c220bacb6	SSH	TCP	22	Custom ▾ <input type="text" value="0.0.0.0/0"/> X	<input type="button" value="Delete"/>
-	All ICMP - IPv4	TCP	80	Anywhere ▾ <input type="text" value="0.0.0.0/0"/> X	<input type="button" value="Delete"/>

[Add rule](#) [Preview changes](#) [Save rules](#)

Add rule and fill all selected information after that save rules

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Make the same changes in the security group of the second EC2 machine, UbuntuPC2. Go to the security settings of UbuntuPC2 and click on the storage group.

The screenshot shows the AWS EC2 Instances page for instance **i-0be5c3462f192d51c (UbuntuPC2)**. The Security tab is selected. A red box highlights the 'Security groups' section, which contains a link to 'Click here'. Another red box highlights the 'sg-007362dba1958ced8 (launch-wizard-3)' security group.

1 EC2 > Instances > i-0be5c3462f192d51c

2 Instance ID: i-0be5c3462f192d51c (UbuntuPC2)

3 Security

4 sg-007362dba1958ced8 (launch-wizard-3)

Click here

Edit the inbound rules of the storage group.

The screenshot shows the AWS EC2 Security Groups page. A specific security group, "sg-007362dba1958ced8 - launch-wizard-3", is selected and highlighted with a red box. The "Inbound rules" tab is active, showing two entries:

Name	Security group rule ID	IP version	Type	Protocol
-	sgr-08807ba7966dcd9df	IPv4	HTTP	TCP
-	sgr-06acde44c220bacb6	IPv4	SSH	TCP

A red box highlights the "Edit inbound rules" button at the top right of the Inbound rules table. The URL in the browser address bar is: ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#SecurityGroup:securitygroup-Details:sg-007362dba1958ced8

To add the rule, select "ALL ICMP-IPv4" as the rule type, choose "Anywhere" under the source, which represents the IPv4 range, and save the rules.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-08807ba7966dcd9df	HTTP	TCP	80	Custom ▾ <input type="text" value="0.0.0.0/0"/> X	Delete
sgr-06acde44c220bacb6	SSH	TCP	22	Custom ▾ <input type="text" value="0.0.0.0/0"/> X	Delete
-	All ICMP - IPv4	ICMP	All	Anywhere-IPv4 ▾ <input type="text" value="0.0.0.0/0"/> X	Delete

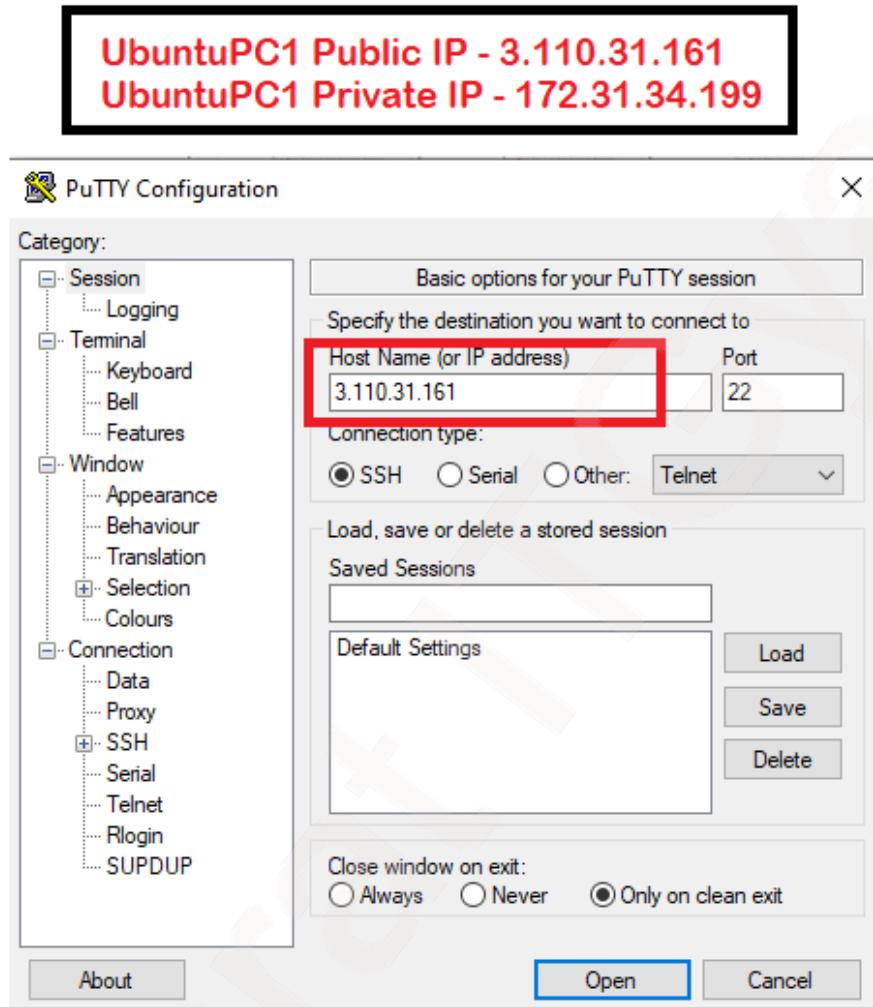
Add rule [Info](#)

Cancel [Preview changes](#) Save rules

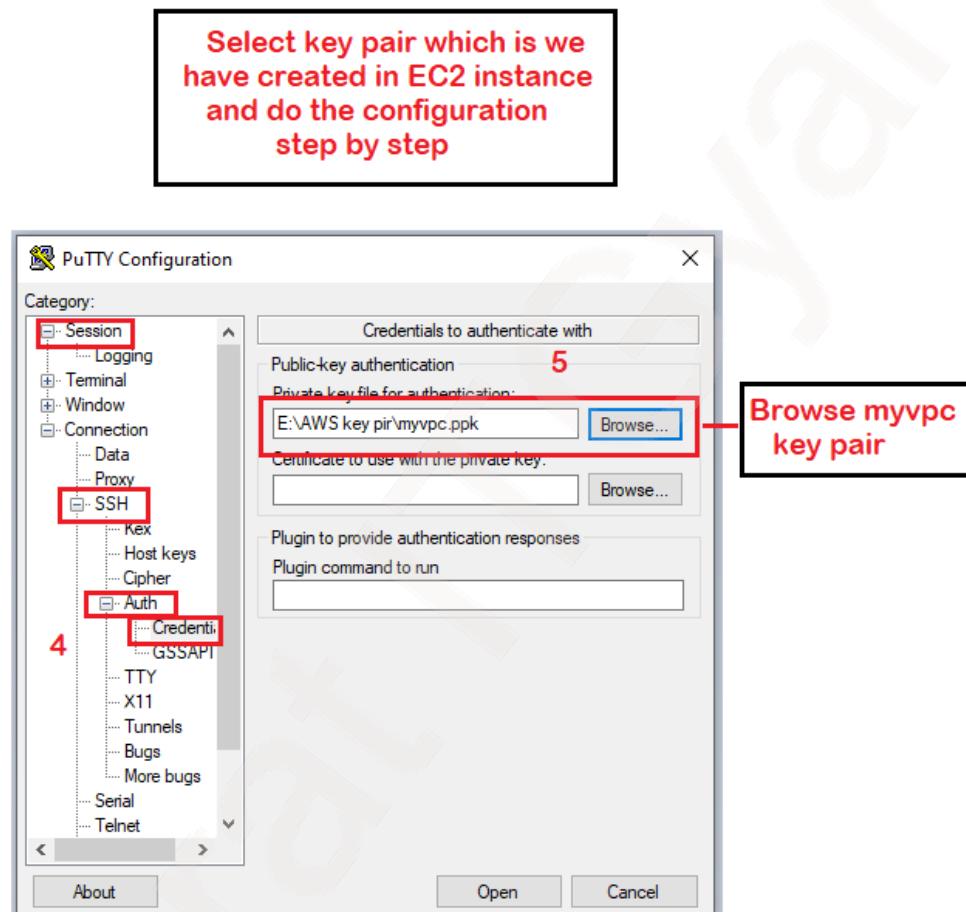
Add rule and fill all information and save rules

We have successfully edited the inbound rules for EC2 instance 1 and 2 machines, enabling the necessary changes to enable pinging between the two machines.

To connect to the EC2 instance using PuTTY, open PuTTY and enter the public IP address of the instance.



Afterward, follow the step-by-step process outlined below. If you have a downloaded .pem file, open PuTTY Key Generator and convert the .pem file to .ppk format.



After that, click on "Open."

To log in to the Ubuntu EC2 instance, use the default login name 'ubuntu.' Enter the login name as 'ubuntu' and press enter. Once logged in, proceed to ping the IP address of UbuntuPC1 from UbuntuPC2.

Ubuntu PC1 IP address 172.31.34.199

Ubuntu PC2 IP address 172.31.43.14

Login and ping EC2 instance - Ubuntu PC2 ip 172.31.43.14 and check Ping is working or not

The screenshot shows two PuTTY windows. The top window is titled "UbuntuPC1 Public IP" and has the address "3.110.31.161" and "login as: ubuntu". The bottom window is titled "UbuntuPC1 Private IP" and has the address "ubuntu@ip-172-31-34-199: ~". A red box highlights the terminal area of the bottom window, and a red arrow points from the text "ping 172.31.43.14" to the output of the ping command, which shows four successful packets sent to the Ubuntu PC2 private IP. A red box also highlights the text "UbuntuPC2 Private IP - 172.31.43.14" at the bottom of the window.

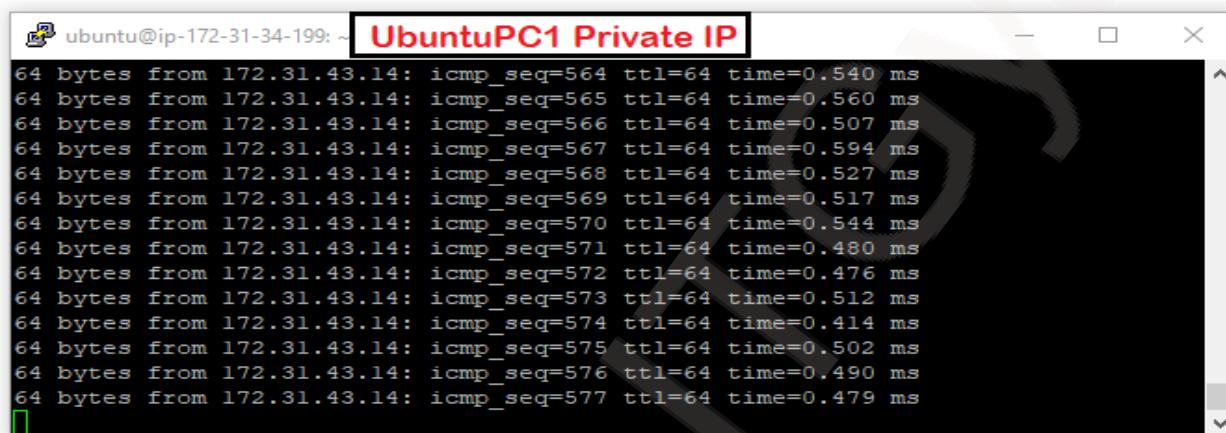
```
ubuntu@ip-172-31-34-199:~$ ping 172.31.43.14
PING 172.31.43.14 (172.31.43.14) 56(84) bytes of data.
64 bytes from 172.31.43.14: icmp_seq=1 ttl=64 time=0.485 ms
64 bytes from 172.31.43.14: icmp_seq=2 ttl=64 time=0.495 ms
64 bytes from 172.31.43.14: icmp_seq=3 ttl=64 time=0.547 ms
64 bytes from 172.31.43.14: icmp_seq=4 ttl=64 time=0.529 ms
```

After the initial access via PuTTY, we need to use PuTTY again to log in to the Ubuntu EC2 instance. We need to follow the same process as described above. Then, access the second UbuntuPC2 via PuTTY and ping both machines to establish connectivity between them.

The default login name for an EC2 Ubuntu machine is 'ubuntu.' Enter the login name as 'ubuntu' and press enter. After logging in, ping the IP address of UbuntuPC2 to UbuntuPC1.

Ubuntu PC1 IP address 172.31.34.199 / Ubuntu PC2 IP address 172.31.43.14

Successfully Ping UbuntuPC1 IP to UbuntuPC2 IP

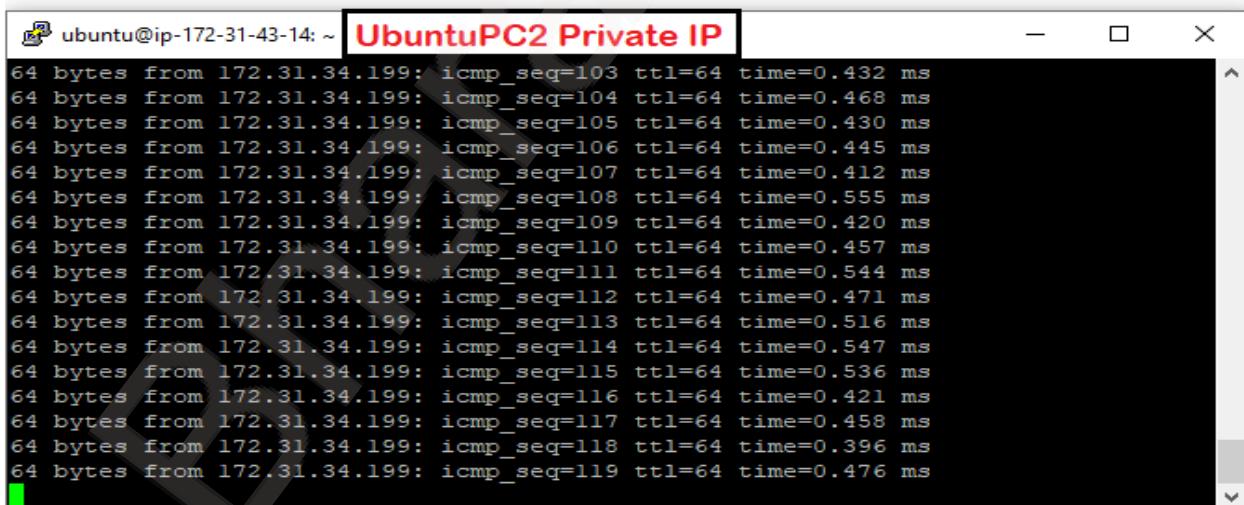


```
ubuntu@ip-172-31-34-199: ~
```

```
UbuntuPC1 Private IP
```

```
64 bytes from 172.31.43.14: icmp_seq=564 ttl=64 time=0.540 ms
64 bytes from 172.31.43.14: icmp_seq=565 ttl=64 time=0.560 ms
64 bytes from 172.31.43.14: icmp_seq=566 ttl=64 time=0.507 ms
64 bytes from 172.31.43.14: icmp_seq=567 ttl=64 time=0.594 ms
64 bytes from 172.31.43.14: icmp_seq=568 ttl=64 time=0.527 ms
64 bytes from 172.31.43.14: icmp_seq=569 ttl=64 time=0.517 ms
64 bytes from 172.31.43.14: icmp_seq=570 ttl=64 time=0.544 ms
64 bytes from 172.31.43.14: icmp_seq=571 ttl=64 time=0.480 ms
64 bytes from 172.31.43.14: icmp_seq=572 ttl=64 time=0.476 ms
64 bytes from 172.31.43.14: icmp_seq=573 ttl=64 time=0.512 ms
64 bytes from 172.31.43.14: icmp_seq=574 ttl=64 time=0.414 ms
64 bytes from 172.31.43.14: icmp_seq=575 ttl=64 time=0.502 ms
64 bytes from 172.31.43.14: icmp_seq=576 ttl=64 time=0.490 ms
64 bytes from 172.31.43.14: icmp_seq=577 ttl=64 time=0.479 ms
```

Successfully Ping UbuntuPC2 IP to UbuntuPC1 IP



```
ubuntu@ip-172-31-43-14: ~
```

```
UbuntuPC2 Private IP
```

```
64 bytes from 172.31.34.199: icmp_seq=103 ttl=64 time=0.432 ms
64 bytes from 172.31.34.199: icmp_seq=104 ttl=64 time=0.468 ms
64 bytes from 172.31.34.199: icmp_seq=105 ttl=64 time=0.430 ms
64 bytes from 172.31.34.199: icmp_seq=106 ttl=64 time=0.445 ms
64 bytes from 172.31.34.199: icmp_seq=107 ttl=64 time=0.412 ms
64 bytes from 172.31.34.199: icmp_seq=108 ttl=64 time=0.555 ms
64 bytes from 172.31.34.199: icmp_seq=109 ttl=64 time=0.420 ms
64 bytes from 172.31.34.199: icmp_seq=110 ttl=64 time=0.457 ms
64 bytes from 172.31.34.199: icmp_seq=111 ttl=64 time=0.544 ms
64 bytes from 172.31.34.199: icmp_seq=112 ttl=64 time=0.471 ms
64 bytes from 172.31.34.199: icmp_seq=113 ttl=64 time=0.516 ms
64 bytes from 172.31.34.199: icmp_seq=114 ttl=64 time=0.547 ms
64 bytes from 172.31.34.199: icmp_seq=115 ttl=64 time=0.536 ms
64 bytes from 172.31.34.199: icmp_seq=116 ttl=64 time=0.421 ms
64 bytes from 172.31.34.199: icmp_seq=117 ttl=64 time=0.458 ms
64 bytes from 172.31.34.199: icmp_seq=118 ttl=64 time=0.396 ms
64 bytes from 172.31.34.199: icmp_seq=119 ttl=64 time=0.476 ms
```

We can ping two machines simultaneously by creating a VPC and utilizing two separate subnets.