

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot displays two consecutive screenshots of the AWS VPC Console interface.

**Screenshot 1: CreateVpc | VPC Console**

This screen shows the "CreateVpc" wizard. The "VPC settings" step is selected. Under "Resources to create", the "VPC only" option is chosen. A "Name tag - optional" field contains "TCS". Under "IPv4 CIDR block", the "IPv4 CIDR manual input" option is selected, and the CIDR block "12.0.0.0/16" is entered. The status bar at the bottom indicates "You successfully created vpc-0e72f9cb13b8569c5 / TCS".

**Screenshot 2: VPC | us-east-1**

This screen shows the "VPC dashboard" for the newly created VPC. The VPC ID is "vpc-0e72f9cb13b8569c5" and it is named "TCS". The "Details" section shows the following configuration:

Setting	Value
VPC ID	vpc-0e72f9cb13b8569c5
State	Available
DNS resolution	Enabled
Main network ACL	acl-05d5aa97c3dfcdde
IPv6 CIDR (Network border group)	-
State	default
Default VPC	No
Network Address Usage metrics	Disabled
Block Public Access	Off
DHCP option set	dopt-059ba96315925ab67
IPv4 CIDR	12.0.0.0/16
Route 53 Resolver DNS Firewall rule groups	-
DNS hostnames	Disabled
Main route table	rtb-07d85c31a80308a58
IPv6 pool	-
Owner ID	221082205626

The status bar at the bottom indicates "You successfully created vpc-0e72f9cb13b8569c5 / TCS".

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the AWS VPC Dashboard. A success message at the top states: "You successfully created **vpn-0e72f9cb13b8569c5 / TCS**". Below this, the "Your VPCs" section lists two VPCs:

Name	VPC ID	State	Block Public...	IPv4 CIDR
-	<a href="#">vpn-0386d07785fc99c9c</a>	Available	Off	172.31.0.0/16
TCS	<a href="#">vpn-0e72f9cb13b8569c5</a>	Available	Off	12.0.0.0/16

A message below the table says "Select a VPC above".

The screenshot shows the AWS EC2 Instances page. The URL is <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances>. The "Name and tags" section has "TCS- LINUX- VPC" entered. The "Application and OS Images (Amazon Machine Image)" section shows a search bar and a "Quick Start" tab selected, with various OS icons like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. The "Summary" section shows "Number of instances": 1, "Software Image (AMI)": "Provided by Red Hat, Inc. ami-0c7af5fe939f2677f", "Virtual server type (instance type)": "t2.micro", and "Firewall (security group)": "New security group".

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot displays three separate windows from the AWS CloudShell interface, illustrating the process of setting up a VPC and launching an EC2 instance.

**Top Window:** Shows the "Launch an instance" wizard. Key settings include:

- Key pair name - required:** TCS-L
- VPC - required:** vpc-0e72f9cb13b8569c5 (TCS) 12.0.0/16
- Subnet:** Select (dropdown menu)
- Auto-assign public IP:** Select (dropdown menu)
- Firewall (security groups):** Create security group (radio button selected)

**Right Panel:** Summary section shows:

- Number of instances: 1
- Software Image (AMI): Provided by Red Hat, Inc. ami-0c7af5fe939f2677f
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group

**Bottom Buttons:** Cancel, Launch instance, Preview code.

**Middle Window:** Shows the "Subnets" page under the VPC dashboard. A success message states: "You have successfully created 1 subnet: subnet-0b54472b8630f74e6". The table lists the subnet details:

Subnet ID	Name	State	VPC	Block Public
subnet-0b54472b8630f74e6	TCS-LINUX	Available	vpc-0e72f9cb13b8569c5   TCS	Off

**Bottom Window:** Shows the "CloudShell" interface with a search bar and various icons.

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the AWS EC2 Launch Instance wizard. On the left, under 'Network settings', a dropdown for 'VPC - required' is set to 'vpc-0e72f9cb13b8569c5 (TCS)'. A 'Subnet' dropdown shows 'subnet-0b54472b8630f74e6 TCS-LINUX'. Under 'Auto-assign public IP', it is set to 'Disable'. In the 'Firewall (security groups)' section, there are two buttons: 'Create security group' (selected) and 'Select existing security group'. On the right, the 'Summary' section shows 'Number of instances' set to 1. It includes details about the 'Software Image (AMI)', 'Virtual server type (instance type)', and 'Firewall (security group)'. Buttons for 'Cancel', 'Launch instance', and 'Preview code' are at the bottom.

**Network settings**

VPC - required: vpc-0e72f9cb13b8569c5 (TCS)  
Subnet: subnet-0b54472b8630f74e6 TCS-LINUX  
Auto-assign public IP: Disable  
Firewall (security groups): Create security group

**Summary**

Number of instances: 1  
Software Image (AMI): Provided by Red Hat, Inc.  
Virtual server type (instance type): t2.micro  
Firewall (security group): New security group

Success

Successfully initiated launch of instance (i-08368e71afac36c78)

**Next Steps**

What would you like to do next with this instance, for example "create alarm" or "create backup"

Create billing and free tier usage alerts  
Connect to your instance  
Connect an RDS database  
Create EBS snapshot policy

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 30°C Mostly cloudy 1:39 PM 12/28/2024

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.

Connect to your instance  
Once your instance is running, log into it from your local computer.  
Connect to instance

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 EBS snapshot policy  
Create a policy that automates the creation, retention, and deletion of EBS snapshots.  
Create EBS snapshot policy

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 30°C Mostly cloudy 1:40 PM 12/28/2024

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot displays two sequential steps in the AWS Cloud Console:

**Step 1: Launch an instance**

The user is creating a new EC2 instance. In the "Name and tags" section, the instance is named "TCS-WINDOWS-VPC". The "Virtual server type (instance type)" is set to "t2.micro". The "Software Image (AMI)" dropdown is open, showing the "Search our full catalog including 1000s of application and OS images" field. The "Launch instance" button is highlighted in orange at the bottom right.

**Step 2: Create subnet**

The user is creating a new subnet named "TCS-WIN". The "Availability Zone" dropdown shows "No preference". The "IPv4 VPC CIDR block" is set to "12.0.0.0/16", which covers "65,536 IPs". The "Tags" section is partially visible at the bottom.

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the AWS CloudShell interface with the following details:

- Key pair name - required:** TCS-WIN
- Network settings:**
  - VPC - required:** vpc-0e72f9cb13b8569c5 (TCS) 12.0.0.0/16
  - Subnet:** subnet-0b54472b8630f74e6 (TCS-LINUX)
  - Auto-assign public IP:** Disable
- Summary:** Number of instances: 1
- Software Image (AMI):** Windows\_Server-2022-English-Full-Base ami-015f002db921fbf07
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 30 GiB
- Launch instance** button

**Success:** Successfully initiated launch of instance (i-0752a0e4c0c561582)

**Launch log:**

**Next Steps:**

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database
- Create EBS snapshot policy

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 1:55 PM 12/28/2024

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The image consists of three vertically stacked screenshots from the AWS Management Console:

- Screenshot 1: EC2 Instances Page**  
Shows the EC2 Instances page with two running t2.micro instances: "TCS-WINDOW..." and "TCS-LINUX-VC".
- Screenshot 2: Putty Configuration Dialog**  
Shows the Putty Configuration dialog for connecting to the instance with ID i-08368e71afac36c78. The "SSH" category is selected, and the "Auth" section shows a private key file path: C:\Users\VIJAY\Downloads\TCS-L.pem.
- Screenshot 3: SSH Connection Details**  
Shows the "Connect to instance" details for the same instance. It includes instructions for opening an SSH client, locating the private key, running chmod 400 on it, and connecting using the private IP 12.0.51.227. An example command is provided: ssh -i "TCS-L.pem" ec2-user@12.0.51.227.

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot displays two browser windows side-by-side, both showing the AWS VPC console interface.

**Top Window (Create VPC Step):**

- Resources to create:** "VPC and more" is selected.
- Name tag auto-generation:** "Auto-generate" is checked, and the tag "WIPRO-VPC" is displayed.
- IPv4 CIDR block:** CIDR 13.0.0.0/16 is specified, covering 65,536 IPs.
- IPv6 CIDR block:** "No IPv6 CIDR block" is selected.

**Middle Window (Create VPC resources Step):**

- Details:**
  - Checklist items:
    - Create VPC: vpc-0444e50b07350539a
    - Enable DNS hostnames
    - Enable DNS resolution
    - Verifying VPC creation: vpc-0444e50b07350539a
    - Create S3 endpoint: vpc-050f96d87212fe05
    - Create subnet: subnet-0bdbad4f9047b5525
    - Create subnet: subnet-0e2c9b4d70dd7138b
  - Subsequent steps:
    - Create subnet
    - Create subnet
    - Create subnet
    - Create subnet
    - Create internet gateway
    - Attach internet gateway to the VPC
    - Create route table
    - Create route
    - Associate route table
    - Associate route table
    - Associate route table
    - Create route table

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

Name	VPC ID	State	Block Public...	IPv4 CIDR
-	ypc-0386d07785fc99c9c	Available	Off	172.31.0.0/16
WIPRO-VPC-vpc	ypc-0444e50b07350539a	Available	Off	13.0.0.0/16

**Launch an instance**

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags**

Name: WIPRO-LINUX-VPC

**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

**Summary**

Number of instances: 1

Software Image (AMI): Provided by Red Hat, Inc. ami-0c7af5fe939f2677f

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

**Actions**

Cancel, Launch instance, Preview code

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The image consists of three vertically stacked screenshots of the AWS VPC Console, illustrating the process of launching a new EC2 instance.

**Screenshot 1: Network Settings - Subnet Creation**

This screenshot shows the "Network settings" section of the "Launch an instance" wizard. It includes fields for selecting a VPC (vpc-0444e50b07350539a), choosing a subnet (subnet-0db78282eb06a2035), and configuring security groups (Create security group selected). The "Auto-assign public IP" and "Firewall (security groups)" sections are also visible. On the right, a summary panel shows 1 instance being launched, using the "Software Image (AMI) amzn-ami-hvm-2024.09.1-x86\_64-v2.0.20240912" and "Virtual server type (instance type) t2.micro". A large orange "Launch instance" button is prominent.

**Screenshot 2: Launch Wizard - Step 1: Set Instance Details**

This screenshot shows the first step of the "Launch an instance" wizard. It asks for a "Name and tags" (WIBRO-WIN-VPC) and provides a search bar for "Application and OS Images (Amazon Machine Image)". Below the search bar are tabs for "AMI from catalog", "Recents", and "Quick Start". The "Summary" panel on the right shows 1 instance being launched with the same AMI and instance type as in the previous screenshot.

**Screenshot 3: Launch Wizard - Step 2: Configure Storage and Networking**

This screenshot shows the second step of the wizard, where users can configure storage volumes and networking. It includes sections for "Storage (volumes)", "Virtual interface (network interface)", and "Associate existing volume". The "Summary" panel on the right remains consistent with 1 instance being launched.

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

**Network settings**

- VPC - required: vpc-0444e50b07350539a (WIPRO-VPC-vpc)
- Subnet: subnet-02c60533388b1c1bd (WIPRO-VPC-subnet-private1-us-east-1a)
- Auto-assign public IP: Disable

**Summary**

- Number of instances: 1
- Software Image (AMI): Windows\_Server-2022-English-Full-Base (ami-015f002db921fb07)
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 30 GiB

**Instances (4)**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
TCS-WINDOWS...	i-0752a0e4c0c561582	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e
TCS-LINUX-VM...	i-08368e71afac36c78	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e
WIBRO-WIN-V...	i-00897e64bf9c6e532	Running	t2.micro	Initializing	<a href="#">View alarms</a> +	us-east-1a
WIBRO-LINUX...	i-06e275cd96c44f851	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1c

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the AWS VPC console interface. A modal dialog box is open, confirming the deletion of 14 resources. The resources listed are:

Name	Resource ID	State
WIPRO-VPN-igw	igw-0ca5ef9a77f36261	Ava
WIPRO-VPN-rtb-private1-us-east-1a	rtb-05e0aa53a99944827	-
WIPRO-VPN-rtb-private2-us-east-1b	rtb-0e48316f006860665	-
WIPRO-VPN-rtb-private3-us-east-1c	rtb-0c0375b27a7d74b22	-
WIPRO-VPN-rtb-public	rtb-024faf88b985b700d	-

To confirm deletion, type `delete` in the field below:

`delete`

Buttons: Cancel, Delete

The screenshot shows the AWS VPC console interface, specifically the 'Create VPC' page. The 'VPC settings' section includes:

- Resources to create:** Info - Create only the VPC resource or the VPC and other networking resources. Selected:  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. Selected:  Auto-generate, Value: CTS-VPC
- IPv4 CIDR block:** Info - Determine the starting IP and the size of your VPC using CIDR notation. Value: 14.0.0.0/16 (65,536 IPs)
- IPv6 CIDR block:** Info - CIDR block size must be between /16 and /28.

The 'Preview' section shows the VPC structure:

- VPC:** Show details - Your AWS virtual network, Name: CTS-VPC-vpc
- Subnets (4):** Subnets within this VPC
  - us-east-1a:** CTS-VPC-subnet-public1-us-east-1a, CTS-VPC-subnet-private1-us-east-1a
  - us-east-1b:** CTS-VPC-subnet-public2-us-east-1b, CTS-VPC-subnet-private2-us-east-1b

Buttons: CloudShell, Feedback

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the AWS VPC console with the URL <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateVpcWizard>. The page displays a list of successful steps in the VPC creation process, including creating the VPC itself and multiple subnets across different availability zones. A 'Success' message is visible at the top left.

**Create VPC workflow**

**Success**

**Details**

- Create VPC: vpc-058e3d460be88482b
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-058e3d460be88482b
- Create S3 endpoint: vpc-0fd373115ad23ef3
- Create subnet: subnet-0ad86647f117c5cd9
- Create subnet: subnet-0353894092d002e64
- Create subnet: subnet-080be8d2e66627beb
- Create subnet: subnet-0632220a2e8ab0f66
- Create subnet: subnet-0f5a7e5a1528dc6fb
- Create subnet: subnet-0b09322d87f0dc06b
- Create subnet: subnet-09774698889e4c204
- Create subnet: subnet-0b3f47d4d20af001
- Create subnet: subnet-02ed3f21230f13b01

**CloudShell Feedback**

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 29°C Haze 3:02 PM 12/28/2024

The screenshot shows the AWS EC2 console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances>. The page is titled 'Launch an instance | EC2 | us-east-1'. It displays a search bar and a list of recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and others. A summary section on the right shows the selection of 2 instances, the software image (Red Hat Enterprise Linux 9), the virtual server type (t2.micro), and the firewall (New security group). Buttons for 'Launch instance' and 'Preview code' are visible.

**Launch an instance | EC2 | us-east-1**

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux De

Amazon Machine Image (AMI)

Red Hat Enterprise Linux 9 (HVM), SSD Volume Type  
ami-0c7af5fe939f2677f (64-bit (x86)) / ami-09373db3d247551a0 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Description  
Red Hat Enterprise Linux version 9 (HVM), EBS General Purpose (SSD) Volume Type

Provided by Red Hat, Inc.

**Summary**

Number of instances | Info  
2

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)  
Provided by Red Hat, Inc.  
ami-0c7af5fe939f2677f

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Cancel Launch instance Preview code

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 29°C Haze 3:04 PM 12/28/2024

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows a browser window with the AWS CloudWatch Metrics console. A green success message at the top states: "Successfully initiated launch of instances (i-0391e099f738cb811, i-08e682b126cd51b1f)". Below this, a "Launch log" section is visible. Under "Next Steps", there are four cards: "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy".

**Success**  
Successfully initiated launch of instances (i-0391e099f738cb811, i-08e682b126cd51b1f)

▶ Launch log

**Next Steps**

What would you like to do next with these instances, for example "create alarm" or "create backup"

1 2 3 4 5 6 >

Create billing and free tier usage alerts  
Connect to your instance  
Connect an RDS database  
Create EBS snapshot policy

**Launch an instance**

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: CTS-WIN-VM

Add additional tags

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.

Search our full catalog including 1000s of application and OS images

AMI from catalog Recents Quick Start

Summary

Number of instances: 1

Software Image (AMI): Windows\_Server-2022-English-Full-Gen2

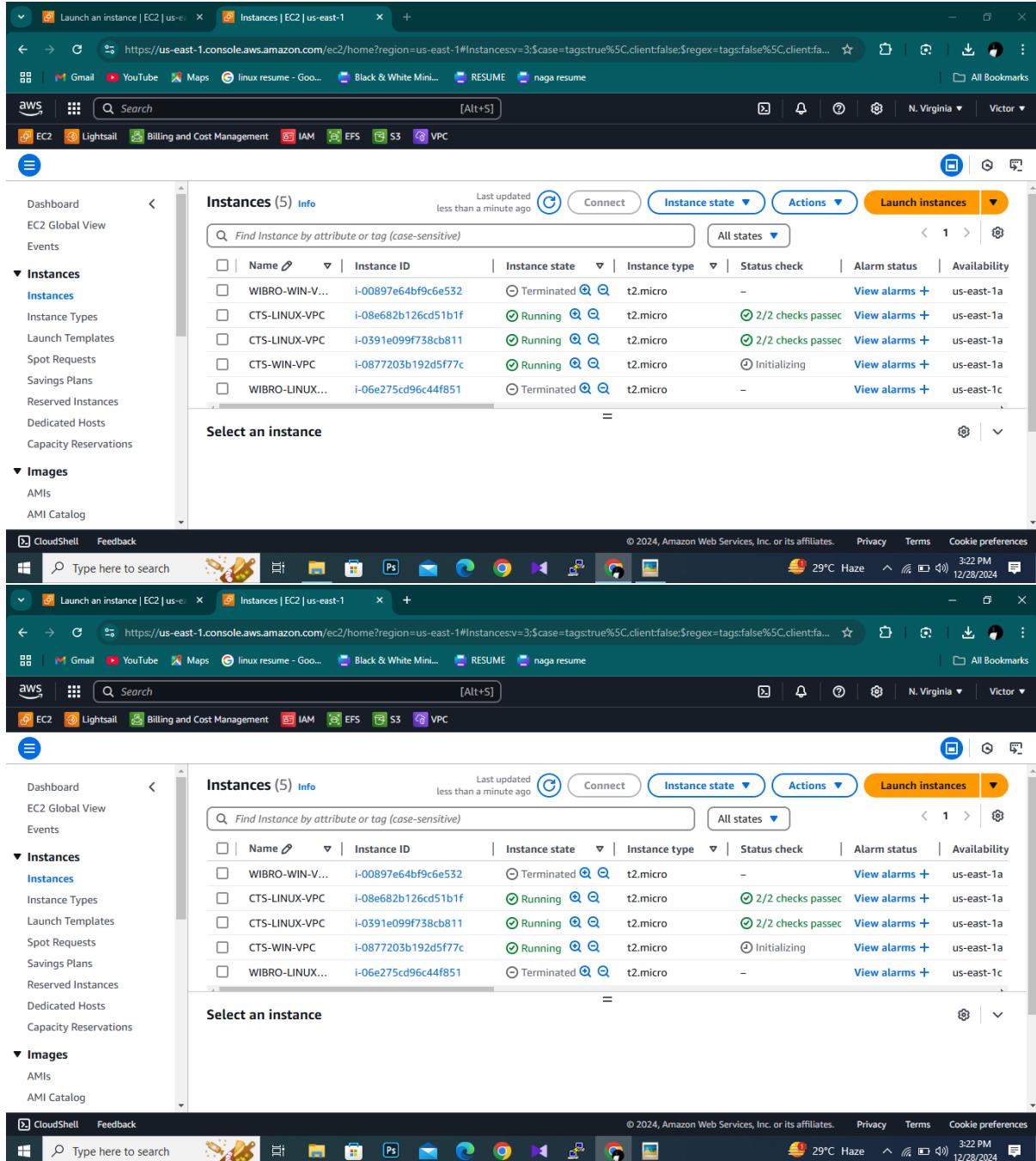
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 30 GiB

Cancel Launch instance

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.



The screenshot shows the AWS Management Console interface for the EC2 service. The top navigation bar includes links for Launch an instance, Instances, and other AWS services like Lightsail, Billing and Cost Management, IAM, EFS, S3, and VPC. The main content area is titled "Instances (5) Info" and displays a table of running EC2 instances. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability. All instances are currently running (t2.micro type) and located in the us-east-1 region. A search bar at the top allows filtering by instance attribute or tag. Below the table, a message says "Select an instance". The left sidebar contains navigation links for Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations), and Images (with sub-links for AMIs and AMI Catalog). The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 12/28/2024 at 3:22 PM.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
WIBRO-WIN-V...	i-00897e64bf9c6e532	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1a
CTS-LINUX-VM	i-08e682b126cd51b1f	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1a
CTS-LINUX-VM	i-0391e099f738cb811	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1a
CTS-WIN-VM	i-0877203b192d5f77c	Running	t2.micro	Initializing	<a href="#">View alarms</a> +	us-east-1a
WIBRO-LINUX...	i-06e275cd96c44f851	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1c

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the AWS VPC Console interface. In the top navigation bar, there are tabs for 'Instances | EC2 | us-east-1' and 'vpcs | VPC Console'. The URL in the address bar is <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs>. The main content area displays 'Your VPCs (1/2) info' with a table showing one entry: 'CTS-VPC-vpc' (VPC ID: `vpc-058e3d460be88482b`, State: Available, Block Public Access: Off, IPv4 CIDR: `14.0.0.0/16`). A modal dialog box titled 'Delete VPC' is open over the table, showing a progress bar at 64% completion with the message 'Deleting subnets...'. Below the progress bar, there is a 'Details' section with a 'Cancel' button and a 'Delete' button. At the bottom of the page, there are tabs for 'Details', 'Resource map', 'CIDRs', 'Flow logs', 'Tags', and 'Integrations'. The 'Details' tab is selected. On the left sidebar, under 'Virtual private cloud', there are links for 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', and 'NAT gateways'. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 12/28/2024 at 3:28 PM.

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the 'Create VPC workflow' page in the AWS VPC console. The progress bar indicates 65% completion. The 'Details' section lists numerous steps completed, including creating the VPC, enabling DNS hostnames and resolution, verifying VPC creation, creating S3 endpoints, subnets, and route tables, and attaching an internet gateway. The 'Associated route table' step is currently active. The AWS navigation bar at the top includes links for EC2, Lightsail, Billing and Cost Management, IAM, EFS, S3, and VPC.

The screenshot shows the 'Create subnet' page in the AWS VPC console. It starts with the 'Associated VPC CIDRs' section, which lists the IPv4 CIDR block 10.0.0.0/16. The 'Subnet settings' section allows specifying the subnet name (task-1), availability zone (No preference), and IPv4 VPC CIDR block (10.0.0.0/16). The AWS navigation bar at the top includes links for EC2, Lightsail, Billing and Cost Management, IAM, EFS, S3, and VPC.

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the AWS VPC console interface. The user is navigating through the 'Subnets' section to 'Create subnet'. In the 'IPv4 VPC CIDR block' field, '10.0.0.0/16' is selected. In the 'IPv4 subnet CIDR block' field, '10.0.0.0/24' is specified, which provides 256 IPs. A tag named 'task-1' is added under 'Tags - optional'. The 'Create subnet' button is visible at the bottom right.

The screenshot shows the AWS EC2 console interface. The user is navigating through the 'Instances' section to 'Launch an instance'. Under 'Name and tags', the instance is named 'task-1'. In the 'Application and OS Images (Amazon Machine Image)' section, 'Red Hat' is selected. The 'Launch instance' button is visible at the bottom right.

Designed and implemented a VPC in alignment with client requirements, allocating specific subnet IP addresses to EC2 instances, ensuring a secure, scalable, and optimized network configuration to meet the client's cloud infrastructure needs.

The screenshot shows the AWS CloudShell interface. At the top, there is a navigation bar with tabs for 'CloudShell' and 'Feedback'. Below the navigation bar, the status bar displays the date and time: '27°C Haze' on the left, 'ENG IN' in the center, and '31-12-2024' on the right. The main area of the CloudShell window contains a terminal session. The terminal output shows the following steps:

```
Using username "ec2-user".
Authenticating with public key "task-1"
Register this system with Red Hat Insights: rhc connect
Example:
  rhc connect --activation-key <key> --organization <org>
The rhc client and Red Hat Insights will enable analytics and additional management capabilities on your system.
View your connected systems at https://console.redhat.com/insights
You can learn more about how to register your system
using rhc at https://red.ht/registration
[ec2-user@ip-10-0-8-177 ~]$
```

Below the terminal, there is a section titled 'EC2 Instance Connect' with the following instructions:

1. Open an SSH port
2. Locate your instance ID
3. Run this command
4. Connect to your instance

Under 'Example:', it shows the command: `ssh -i "task-1.pem" ec2-user@ec2-50-19-143-79.compute-1.amazonaws.com`. A note below states: 'Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.'