

## CLOUD COMPUTING

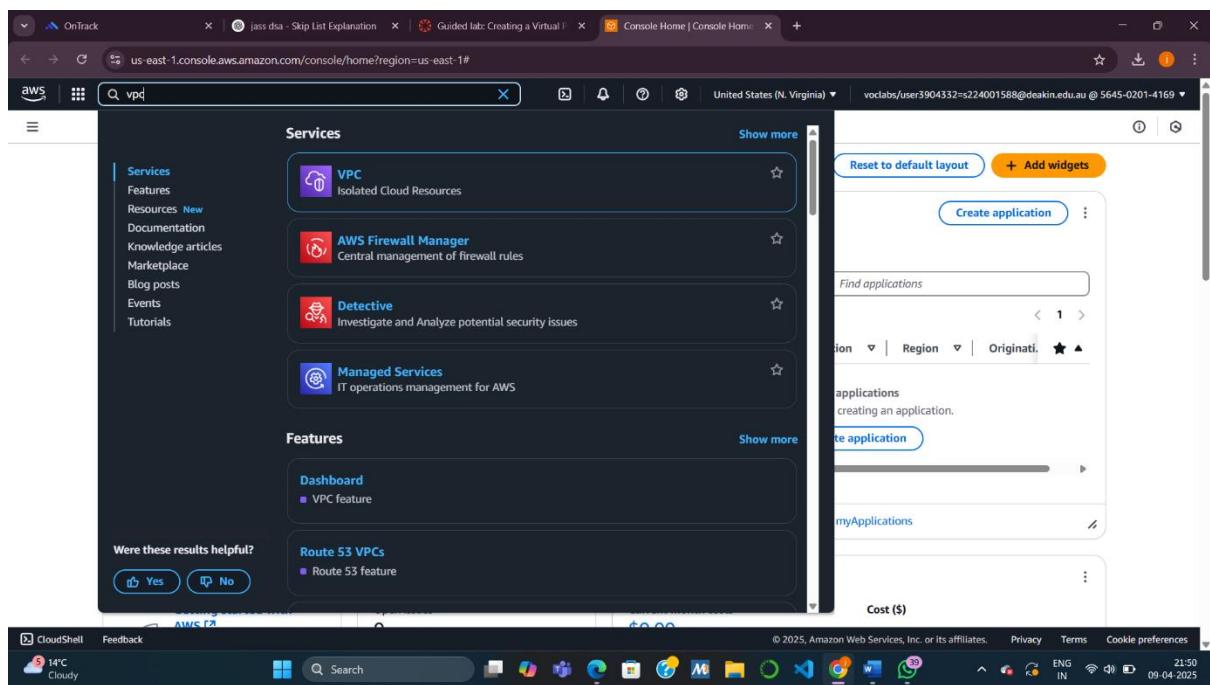
### TASK 5.1P

#### Creating a Virtual Private Cloud

##### Task 1: Creating a VPC

A **VPC (Virtual Private Cloud)** is your own isolated network within AWS.

- **Why:** It allows you to deploy AWS resources in a secure and customizable environment.
- **What you did:**
  - Created a VPC with CIDR block 10.0.0.0/16 (65,536 IPs).
  - Enabled DNS hostnames for EC2 instances for easier access using domain names.



The screenshot shows the AWS VPC Dashboard for the 'us-east-1' region. The left sidebar includes sections for 'Virtual private cloud' (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), 'Security' (Network ACLs, Security groups), and navigation links (EC2 Global View, Filter by VPC). The main content area displays 'Resources by Region' with the following details:

Resource Type	Region	Count
VPCs	United States	1
Subnets	United States	6
Route Tables	United States	1
Internet Gateways	United States	1
Egress-only Internet Gateways	United States	0
NAT Gateways	United States	0
VPC Peering Connections	United States	0
Network ACLs	United States	0
Security Groups	United States	0
Customer Gateways	United States	0

On the right side, there are boxes for 'Service Health' (View complete service health details), 'Settings' (Block Public Access, Zones, Console Experiments), 'Additional Information' (VPC Documentation, All VPC Resources, Forums, Report an Issue), and 'AWS Network Manager' (Provides tools and features to help you manage and...). The bottom of the screen shows the browser's address bar, system tray icons, and a status bar indicating the date and time.

The screenshot shows two windows side-by-side. The left window displays the 'Your VPCs' list in the AWS VPC console. It shows one VPC entry:

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
-	vpc-05c340091899186c5	Available	Off	172.31.0.0/16	-

The right window shows the 'Create VPC' wizard. The first step, 'Create VPC', is selected. The 'VPC settings' section includes:

- Resources to create:** VPC only (selected)
- Name tag - optional:** Lab VPC
- IPv4 CIDR block:** 10.0.0.0/16
- IPv6 CIDR block:** No IPv6 CIDR block selected

Both windows have a standard Windows taskbar at the bottom.

The screenshot shows two consecutive screenshots of the AWS VPC console.

**Screenshot 1: Create VPC Wizard**

This screenshot shows the "Create VPC" wizard. The "VPC settings" step is selected. The "Resources to create" section has "VPC only" selected. The "Name tag" field contains "Lab VPC". Under "IPv4 CIDR block", "IPv4 CIDR manual input" is selected with the value "10.0.0.0/16". Under "IPv6 CIDR block", "No IPv6 CIDR block" is selected. A success message at the bottom states: "You successfully created vpc-0a6e43ba91c66bd16 / Lab VPC".

**Screenshot 2: VPC Dashboard**

This screenshot shows the "VPC dashboard" for the newly created VPC. The VPC ID is "vpc-0a6e43ba91c66bd16". The "Details" tab is selected, showing the following configuration:

- VPC ID:** vpc-0a6e43ba91c66bd16
- State:** Available
- Block Public Access:** Off
- DNS hostnames:** Disabled
- Tenancy:** default
- DHCP option set:** dopt-081e982ce1d379c8d
- Main network ACL:** acl-011a580d62c9d58ef
- Default VPC:** No
- IPv4 CIDR:** 10.0.0.0/16
- IPv6 pool:** -
- Network Address Usage metrics:** Disabled
- Route 53 Resolver DNS Firewall rule groups:** -
- Owner ID:** 564502014169

The "Resource map" tab is also visible, showing the VPC, Subnets (0), Route tables (1), and Network ACLs (0).

The screenshot shows two consecutive pages from the AWS VPC console.

**Page 1: Your VPCs (1) Info**

- Shows a table with one item:
 

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
-	vpc-05c340091899186c5	Available	Off	172.31.0.0/16	-
- Actions button: Create VPC
- CloudShell and Feedback buttons at the bottom.

**Page 2: Edit VPC settings**

- VPC details:**
  - VPC ID: vpc-0a6e43ba91c66bd16
  - Name: Lab VPC
- DHCP settings:**
  - DHCP option set: dopt-081e982ce1d379c8d
- DNS settings:**
  - Enable DNS resolution: checked
  - Enable DNS hostnames: unchecked
- Network Address Usage metrics settings:**
  - Enable Network Address Usage metrics: unchecked
- Buttons: Cancel, Save
- CloudShell and Feedback buttons at the bottom.

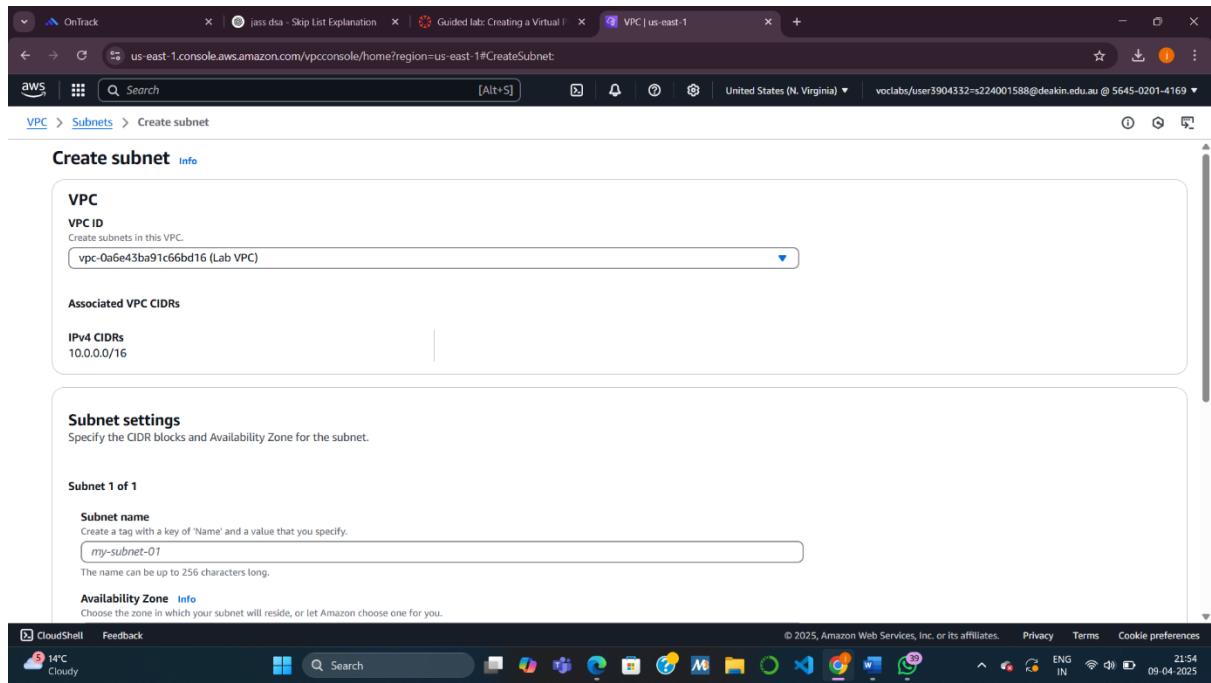
## Task 2: Creating Subnets

Subnets divide your VPC into smaller segments.

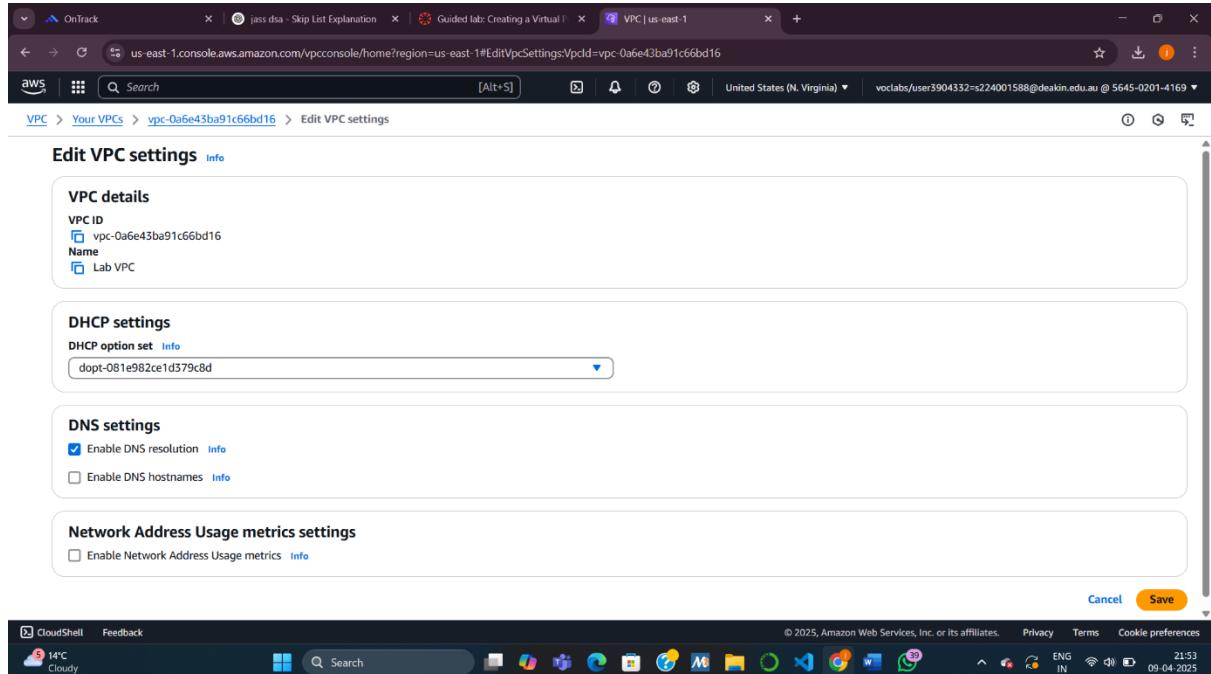
### 2.1 Creating a Public Subnet

- Why:** Public subnets are for resources that need internet access (e.g., web servers).
- What you did:**
  - Created a subnet with CIDR block 10.0.0.0/24 (256 IPs).

- Enabled auto-assign public IP so instances launched here get internet access.



The screenshot shows the AWS VPC console interface. The user is creating a new subnet under a specific VPC. The VPC ID is listed as `vpc-0a6e43ba91c66bd16 (Lab VPC)`. An IPv4 CIDR block of `10.0.0.0/16` is selected. In the **Subnet settings** section, a subnet named `my-subnet-01` is being created, which can be up to 256 characters long. The availability zone is set to `Choose the zone in which your subnet will reside, or let Amazon choose one for you.`. The browser status bar indicates the URL is `us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet`.

The screenshot shows the AWS VPC console interface. The user is editing the settings for a specific VPC, identified by its ID `vpc-0a6e43ba91c66bd16`. The **VPC details** section shows the VPC ID and name (`Lab VPC`). Under **DNS settings**, the `Enable DNS resolution` checkbox is checked. In the **Network Address Usage metrics settings** section, the `Enable Network Address Usage metrics` checkbox is unchecked. The browser status bar indicates the URL is `us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditVpcSettings?vpcId=vpc-0a6e43ba91c66bd16`.

The screenshot shows the AWS VPC Subnets console. A success message at the top states: "You have successfully created 1 subnet: subnet-06ca327af856ee60e". The main table displays one subnet entry:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
Public Subnet	subnet-06ca327af856ee60e	Available	vpc-0a6e43ba91c66bd16   Lab...	Off	10.0.0.0/24

The left sidebar shows navigation options for VPC dashboard, EC2 Global View, Virtual private cloud (Subnets selected), Security, and CloudShell.

## 2.2 Creating a Private Subnet

- Why:** Private subnets are for internal resources (e.g., databases).
- What you did:**
  - Created a subnet with CIDR block 10.0.2.0/23 (512 IPs).
  - This subnet has no public access by default.

The screenshot shows two views of the AWS VPC console. The top view is the 'Edit subnet settings' page for a subnet named 'Public Subnet'. It includes sections for 'Auto-assign IP settings' (with 'Enable auto-assign public IPv4 address' checked), 'Resource-based name (RBN) settings' (with 'IP name' selected), and 'Hostname type' (with 'IP name' selected). The bottom view is the 'Subnets' dashboard, showing a table of subnets. One row is selected for 'Public Subnet' (subnet ID: subnet-06ca327af856ee60e), which is listed under the 'Available' state and associated with a specific VPC and CIDR range.

Subnet	Subnet ID	Name	VPC	Block Public Access	IPv4 CIDR
Public Subnet	subnet-06ca327af856ee60e	Public Subnet	vpc-0a6e43ba91c66bd16   Lab...	Off	10.0.0.0/24

The screenshot shows the 'Create subnet' wizard on the AWS VPC console. The first step, 'Subnet 1 of 1', is displayed. In the 'VPC ID' section, 'vpc-0a6e43ba91c66bd16 (Lab VPC)' is selected. Under 'Associated VPC CIDRs', 'IPv4 CIDRs' is set to '10.0.0.0/16'. The 'Subnet settings' section is expanded, showing the 'Subnet name' field containing 'Private Subnet', the 'Availability Zone' set to 'United States (N. Virginia) / us-east-1', and the 'IPv4 VPC CIDR block' set to '10.0.0.0/16'. The 'IPv4 subnet CIDR block' dropdown shows '10.0.2.0/23'. A 'Tags - optional' section is present with a single tag 'Name: Private Subnet'. The browser status bar at the bottom indicates the URL is 'us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet'.

The screenshot shows the AWS VPC Subnets console. A success message at the top states: "You have successfully created 1 subnet: subnet-06b4ff777176be47b". The main table displays one subnet entry:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
Private Subnet	subnet-06b4ff777176be47b	Available	vpc-0a6e43ba91c66bd16   Lab...	Off	10.0.2.0/23

The left sidebar includes sections for VPC dashboard, EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), and Security (Network ACLs, Security groups). The bottom navigation bar shows CloudShell, Feedback, and various system icons.

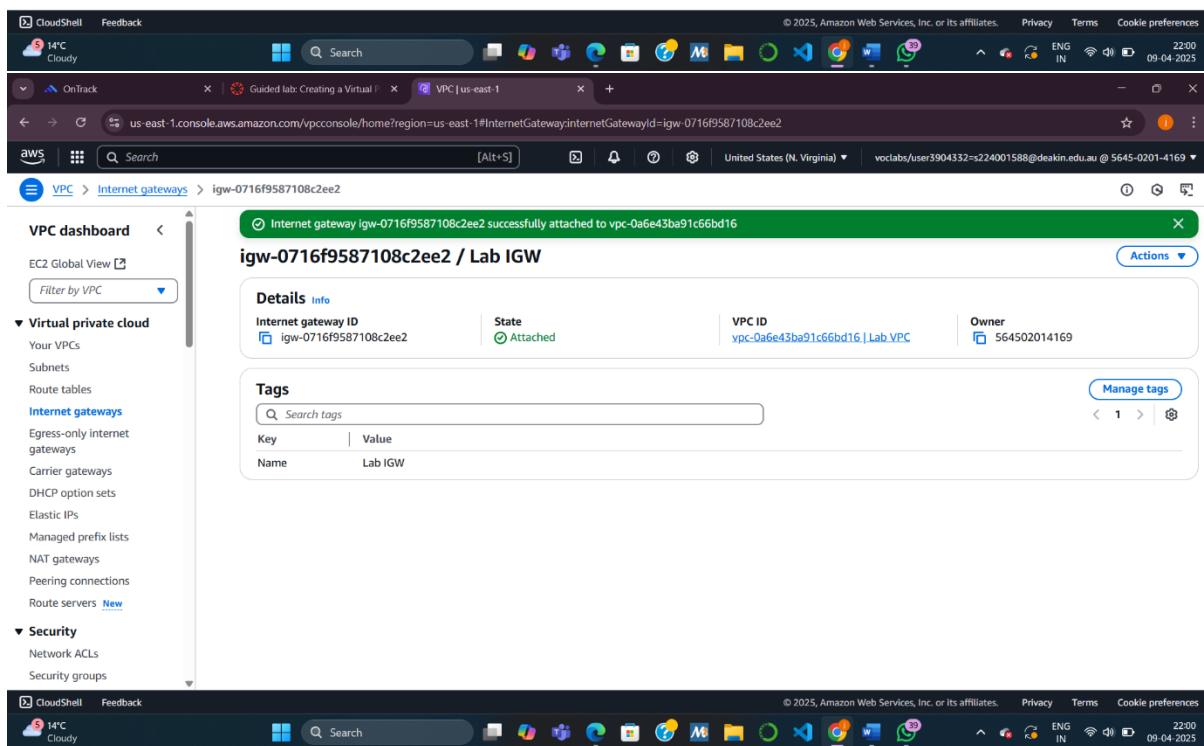
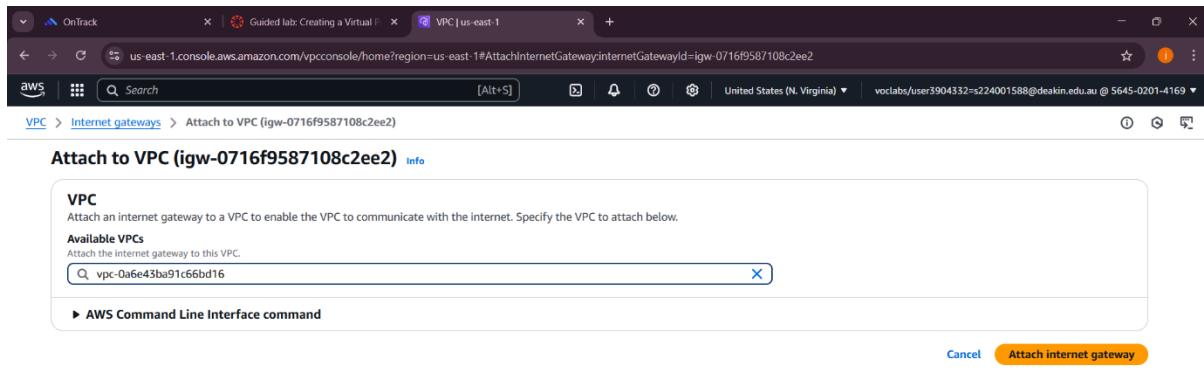
### Task 3: Creating an Internet Gateway

An **Internet Gateway (IGW)** connects your VPC to the internet.

- **Why:** Required for instances in the public subnet to access the internet.
- **What you did:**
  - Created and attached the IGW to your VPC.

The screenshot shows the 'Create internet gateway' wizard in the AWS VPC console. The 'Internet gateway settings' section is displayed, with a 'Name tag' field containing 'Lab IGW'. Below it, the 'Tags - optional' section shows a single tag 'Name: Lab IGW'. At the bottom right are 'Cancel' and 'Create internet gateway' buttons.

The screenshot shows the AWS VPC dashboard. On the left, the navigation menu includes 'Virtual private cloud' and 'Internet gateways'. The main area displays the 'igw-0716f9587108c2ee2 / Lab IGW' details. The 'Details' section shows the Internet gateway ID as 'igw-0716f9587108c2ee2', state as 'Detached', and VPC ID as '-'. The 'Owner' is listed as '564502014169'. The 'Tags' section lists a single tag 'Name: Lab IGW'. A green notification bar at the top states: 'The following Internet gateway was created: igw-0716f9587108c2ee2 - Lab IGW. You can now attach to a VPC to enable the VPC to communicate with the Internet.' An 'Actions' button is also visible.



## Task 4: Configuring Route Tables

Route tables define how traffic moves within and outside your VPC.

### 4.1 Created Private Route Table

- Why:** Handles internal communication within VPC.
- What I did:** Renamed the default route table to “Private Route Table”.

The screenshot shows the AWS VPC Route Tables page. On the left, there's a navigation sidebar with sections like VPC dashboard, Virtual private cloud, Security, and CloudShell. The main area displays a table of route tables:

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-06fc0e01e2a12f5ba	-	-	Yes	vpc-0a6e43ba91c66bd16   Lab
-	rtb-0b5fed6b5d87cf16	-	-	Yes	vpc-05c340091899186c5

**Select a route table**

The screenshot shows the same VPC Route Tables page, but now a new route table is being created. A modal dialog is open over the table, showing the 'Edit Name' field populated with 'Private Route Table'. There are 'Cancel' and 'Save' buttons at the bottom of the dialog.

**rtb-06fc0e01e2a12f5ba**

**Details**

Route table ID	rtb-06fc0e01e2a12f5ba	Main	Yes
VPC	Lab	Owner ID	deakin.edu.au

**Explicit subnet associations**

**Edge associations**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 09-04-2025

## 4.2 Created Public Route Table

- **Why:** Allows internet-bound traffic from the public subnet.
- **What I did:**
  - Created a new route table.
  - Added a route (0.0.0.0/0) to the IGW.
  - Associated this table with the public subnet.

The screenshot shows the 'Create route table' configuration page in the AWS VPC console. It includes sections for 'Route table settings' (with 'Name' set to 'Public Route Table' and 'VPC' set to 'vpc-0a6e43ba91c66bd16 (Lab VPC)'), 'Tags' (with one tag 'Name' set to 'Public Route Table'), and a 'Create route table' button.

**Create route table** Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

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

**VPC**  
The VPC to use for this route table.

**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**  **Value** - *optional*     
You can add 49 more tags.



Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

**Add route** **Remove** **Cancel** **Preview** **Save changes**

Available subnets (1/2)				
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> Public Subnet	subnet-06ca327af856ee60e	10.0.0.0/24	-	Main (rtb-06fc0e01e2a12f5ba / Privat...)
<input type="checkbox"/> Private Subnet	subnet-06b4ff777176be47b	10.0.2.0/23	-	Main (rtb-06fc0e01e2a12f5ba / Privat...)

**Selected subnets**  
subnet-06ca327af856ee60e / Public Subnet **X**

**Cancel** **Save associations**

The screenshot shows the AWS VPC console interface. The main title is "rtb-0f8c296b729418261 / Public Route Table". A green success message at the top states: "You have successfully updated subnet associations for rtb-0f8c296b729418261 / Public Route Table." Below the title, there are four tabs: "Details", "Info", "Routes", and "Subnet associations". The "Routes" tab is selected, showing two routes:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0716f9587108c2ee2	Active	No
10.0.0.0/16	local	Active	No

The left sidebar includes sections for "Virtual private cloud" (Your VPCs, Subnets, Route tables), "Security" (Network ACLs, Security groups), and "CloudShell" (Feedback). The bottom status bar shows the date (09-04-2025) and time (22:05).

## Task 5: Creating a Security Group for the App Server

A **security group** is a virtual firewall for EC2 instances.

- **Why:** To allow HTTP (web) access to your server from anywhere.
- **What I did:**
  - Created a security group called App-SG.
  - Added an inbound rule to allow HTTP traffic from 0.0.0.0/0.

The screenshot shows the 'Create security group' page in the AWS VPC console. In the 'Basic details' section, the security group name is 'App-SG' and the description is 'Allow HTTP traffic'. The VPC selected is 'vpc-0a6e43ba91c66bd16 (Lab VPC)'. In the 'Inbound rules' section, there are no rules listed, but a 'Add rule' button is visible.

**Inbound rules info**  
This security group has no inbound rules.

**Add rule**

The screenshot shows the 'Create security group' page with an inbound rule being added. The rule type is 'HTTP', protocol is 'TCP', port range is '80', source is 'Anyw...', and the description is '0.0.0.0/0'. A warning message at the bottom states: '⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' A 'Delete' button is also visible.

**Inbound rules info**

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	Anyw...	0.0.0.0/0

**Add rule**

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

The screenshot shows the 'Create security group' page with an outbound rule being added. The rule type is 'All traffic', protocol is 'All', port range is 'All', destination is 'Custom', and the description is '0.0.0.0/0'. A 'Delete' button is also visible.

**Outbound rules info**

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

The screenshot shows the AWS VPC console interface. A green success message at the top states: "Security group (sg-066efce02d5564275 | App-SG) was created successfully". Below this, the security group details are shown: Security group name (App-SG), Security group ID (sg-066efce02d5564275), Description (Allow HTTP traffic), Owner (564502014169), VPC ID (vpc-0a6e43ba91c66bd16). The Inbound rules section displays one rule: Name (sgr-0fc4a9704088456de), IP version (IPv4), Type (HTTP), Protocol (TCP), Port range (80). The Outbound rules count is 1. The interface includes a sidebar for VPC dashboard, EC2 Global View, and various cloud services like Subnets, Route tables, Internet gateways, Egress-only Internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, and Route servers. The bottom navigation bar includes CloudShell, Feedback, and a weather widget showing 14°C Cloudy.

## Task 6: Launching an Application Server

You launched an **EC2 instance** (virtual server) in the public subnet.

- **Why:** To test your VPC setup and confirm the web app is accessible.
- **What I did:**
  - Chose the Amazon Linux 2023 AMI.
  - Used subnet: Public Subnet.
  - Applied the App-SG security group.
  - Used a startup script to install a web app and Apache web server.
  - Accessed the app using the **Public IPv4 DNS** of the instance.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, and the main content area displays the following:

- Instances Info:** Last updated less than a minute ago. Buttons for Connect, Instance state, Actions, and Launch instances.
- Search bar:** Find Instance by attribute or tag (case-sensitive).
- Filter options:** Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IP.
- Message:** No instances. You do not have any instances in this region.
- Buttons:** Launch instances.
- Select an instance:** A dropdown menu.

The screenshot shows the AWS EC2 Launch an instance page. The left sidebar is collapsed, and the main content area displays the following steps:

- Name and tags:** Input field for Name (App Server) and Add additional tags button.
- Application and OS Images (Amazon Machine Image):** Search bar for AMIs.
- Quick Start:** Grid of AMI icons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian.
- Summary:** Number of instances (1), Software Image (AMI) (Amazon Linux 2023 AMI 2023.7.2...), Virtual server type (t2.micro), Firewall (New security group), Storage (volumes) (1 volume(s) - 8 GiB).
- Free tier information:** In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t3 micro isn't available).
  - Cancel** button.
  - Launch instance** button.
  - Preview code** button.

The screenshot shows the AWS EC2 'Launch an instance' wizard. The current step is 'Configure storage'. The configuration includes:

- Root volume:** 8 GiB gp3, Root volume, 3000 IOPS, Not encrypted.
- Advanced:** A link to advanced storage options.

On the right side of the screen, there is a summary panel and a 'Free tier' information box. The summary panel shows 1 instance selected. The 'Free tier' box provides details about the free tier benefits for AWS accounts.

At the bottom of the screen, the Windows taskbar is visible with various pinned icons and the system clock showing 22:11 on 09-04-2025.

The screenshot shows the AWS CloudShell interface with two tabs open: "OnTrack" and "Launch an instance | EC2 | us-east-1". The "Launch an instance" tab is active, displaying the "Launch an instance" wizard.

**Allow tags in metadata:** A dropdown menu showing "Select".

**User data - optional:** A text area containing a base64-encoded shell script. The script installs Apache, PHP, MySQL, and MariaDB, then configures them and starts an inventory application.

```
#!/bin/bash
# Install Apache Web Server and PHP
dnf install -y httpd wget php-fpm php-mysql php-json php php-devel
dnf install -y mariadb105-server
# Download Lab files
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACACAD-3-113230/06-lab-mod7-guided-VPC/s3/scripts/al2023-inventory-app.zip -O inventory-app.zip
unzip inventory-app.zip -d /var/www/html/
# Download and install the AWS SDK for PHP
wget https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.zip
unzip aws.zip -d /var/www/html
# Turn on web server
systemctl enable httpd
systemctl start httpd
```

User data has already been base64 encoded

**Summary:**

- Number of instances: 1
- Firewall (security group): App-SG
- Storage (volumes): 1 volume(s) - 8 GiB

**Free tier:** In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

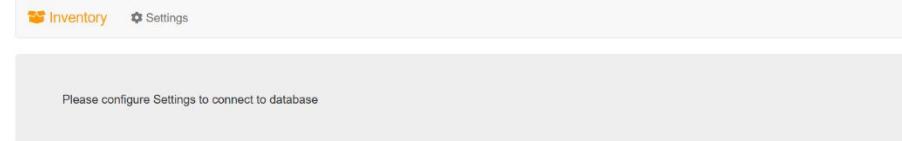
**Buttons:** Cancel, Launch instance, Preview code

**Success message:** Successfully initiated launch of instance (i-0ee1a933edb721888)

**Next Steps:**

- 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
- Manage CloudWatch alarms

The screenshot shows the AWS EC2 Instance Details page for instance `i-0d753516a25254ee4`. The instance is an `t2.micro` App Server running in the us-east-1 region. It has a public IPv4 address of `174.129.178.159` and a private IP DNS name of `ip-10-0-0-229.ec2.internal`. The instance is currently running. The VPC ID is `vpc-0dff13f953b8e95ee`, and it is associated with a public subnet `subnet-0b3ac0fbf158c2e27`. The instance ARN is `arnaws:ec2:us-east-1:015525208414:instance/i-0d753516a25254ee4`. The instance is managed by the user.



**Guided lab: Creating a Virtual Private Cloud**

**Instance summary for i-0d753516a25254ee4 (App Server)**

Updated 3 minutes ago

<b>Instance ID</b> i-0d753516a25254ee4	<b>Public IPv4 address</b> 174.129.178.159   open address	<b>Private IPv4 addresses</b> Public IPv4 DNS copied ec2-174-129-178-159.compute-1.amazonaws.com   open address
<b>IPv6 address</b> -	<b>Instance state</b> Running	<b>Elastic IP addresses</b> -
<b>Hostname type</b> IP name: ip-10-0-0-229.ec2.internal	<b>Private IP DNS name (IPv4 only)</b> ip-10-0-0-229.ec2.internal	<b>AWS Compute Optimizer finding</b> Opt-in to AWS Compute Optimizer for recommendations.   Learn more
<b>Answer private resource DNS name</b> -	<b>Instance type</b> t2.micro	<b>Auto Scaling Group name</b> -
<b>Auto-assigned IP address</b> 174.129.178.159 [Public IP]	<b>VPC ID</b> vpc-0dff13f953b8e95ee (Lab VPC)	<b>Managed</b> false
<b>IAM Role</b> -	<b>Subnet ID</b> subnet-0b3ac0fbf158c2e27 (Public Subnet)	
<b>IMDSv2</b> Required	<b>Instance ARN</b> arnaws:ec2:us-east-1:015525208414:instance/i-0d753516a25254ee4	

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**ACAv3EN-US-LTI... > Assignments > Guided lab: Creating a Virtual Private Cloud**

**Guided lab: Creating a Virtual Private Cloud**

Due No Due Date Points 56 Submitting an external tool

55. Select App Server.

56. From the Details tab, copy the Public IPv4 DNS value.

57. Open a new web browser tab, and enter this public IPv4 DNS value.

If you configured the VPC correctly, the Inventory application and this message should appear: Please configure Settings to connect to database. The appearance of the Inventory application demonstrates that the public subnet was correctly configured.

If the Inventory application does not appear, wait 60 seconds and refresh the page to try again. It can take a couple of minutes for the EC2 instance to boot and run the script that installs the software.

Total score 50/50

- [Task 1] Lab VPC CIDR
- [Task 2A] Public Subnet
- [Task 2B] Auto-Assign Public IP
- [Task 2C] Private Subnet CIDR
- [Task 3] Lab VPC has IGW

Submission  
Apr 9 at 12:22pm  
Submission Details  
Grade: 50.400000000000006 (56 pts possible)  
Graded Anonymously: no  
Comments: No Comments

The screenshot shows the AWS Academy interface with a guided lab titled "Creating a Virtual Private Cloud". The submission report window is open, showing a log of tests:

```

Testing report - Public Subnet has auto-assign public ip enabled.
Testing report - Private Subnet was found and has the correct CIDR.
Testing report - Lab VPC has IGW attached.
Testing report - Found route table named Private Route Table for Lab VPC.
Testing report - Found route table named Public Route Table associated with Public IP.
Testing report - VPC was created manually.
Testing report - Security Group,App-SG, was found for Lab VPC.
Testing report - The application server is reachable.

```

Comments: No Comments

## KNOWLEDGE CHECK MODULE 7

The screenshot shows the AWS Academy interface with a knowledge check for Module 7. The result is displayed as follows:

**KEYBOARD NAVIGATION**

**Knowledge check results**

Your score: 100% (100 points)  
Required score: 70% (70 points)

**Result:** Congratulations! You have completed this knowledge check.

To continue, choose Next in the lower-right corner.

- . What is the difference between an Internet Gateway and a NAT Gateway?
- Internet Gateway (IGW):
  - Allows instances with public IPs in a public subnet to access the internet and be accessed from the internet.
  - Used for inbound and outbound internet traffic.
- NAT Gateway:
  - Allows instances without public IPs in a private subnet to access the internet but blocks inbound traffic from the internet.

Used for outbound-only access from private subnets.

- . How do you make a subnet public?

To make a subnet public:

- Attach an Internet Gateway to the VPC.
- Create or update a route table with a route to 0.0.0.0/0 (internet) via the Internet Gateway.
- Associate the route table with the subnet.
- Enable auto-assign public IP for instances launched in that subnet.
-