

**Project in AWS
Practice Lab**

Build Solutions across VPCs with Peering

Andra-Diana Popescu

2025

ABOUT THIS LAB

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. In this lab, you will create a new VPC for your WordPress blog to run from. You will then create a VPC peering connection between the new VPC and an existing database VPC. By the end of this lab, you will understand how to create a new VPC from scratch, attach internet gateways, edit routing tables, and peer multiple VPCs together.

LEARNING OBJECTIVES

- Create Web_VPC Subnets and Attach a New Internet Gateway
- Create a Peering Connection
- Create an EC2 Instance and configure Wordpress
- Modify the RDS Security Groups to Allow Connections from the Web_VPC VPC
- Test WordPress

AWS Documentation about VPC and subnets:

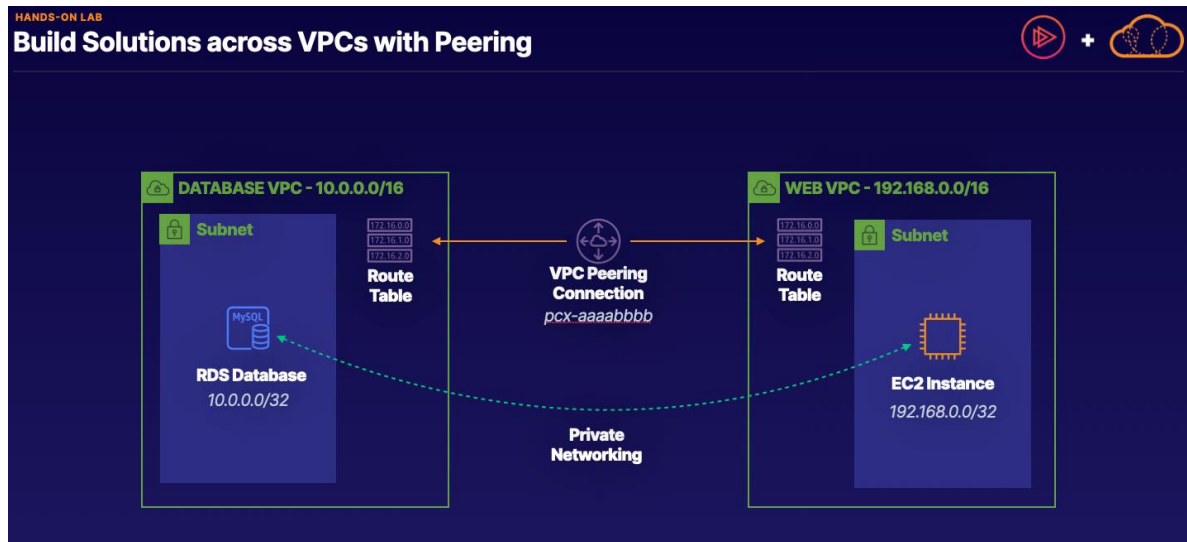
<https://docs.aws.amazon.com/vpc/latest/userguide/configure-subnets.html>

Source: <https://learn.acloud.guru/course/certified-solutions-architect-associate/>

Table of Contents

Lab Diagrams.....	4
Log in to your AWS account	5
1. Create Web_VPC Subnets and Attach a New Internet Gateway	5
1.1. Create a VPC.....	5
1.2. Create a Subnet	7
1.3. Create an Internet Gateway.....	8
2. Create a Peering Connection.....	10
3. Create an EC2 Instance and Configure WordPress.....	14
4. Modify the RDS Security Groups to Allow Connections from the Web_VPC VPC	21

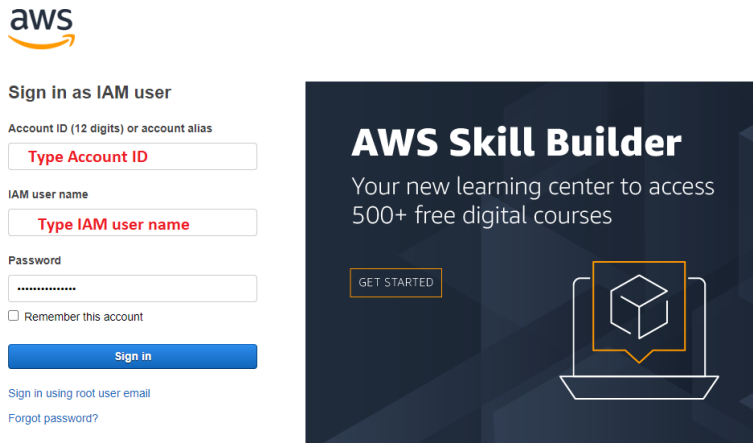
Lab Diagrams



We have the AWS account in **us-east-1** Region, and inside our AWS account, we have a VPC that we'll call it DATABASE VPC. Inside that, we have a subnet and a route table and there's also an RDS MySQL database, which will be created it automatically inside the subnet as well.

Our scenario is that our company is looking to set up a WordPress blog as part of a new internal site. Your database administrator created this RDS database in the VPC that was dedicated to only hosting databases. We need to set up a new WEB VPC with a subnet and routing table, and in the subnet, we'll have an EC2 instance for our blog. Then, we'll configure a VPC peering connection between the 2 VPCs to ensure private networking connection is possible between the EC2 instance and the RDS database.

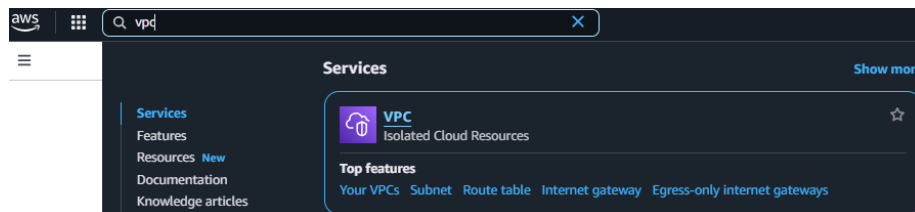
Log in to your AWS account



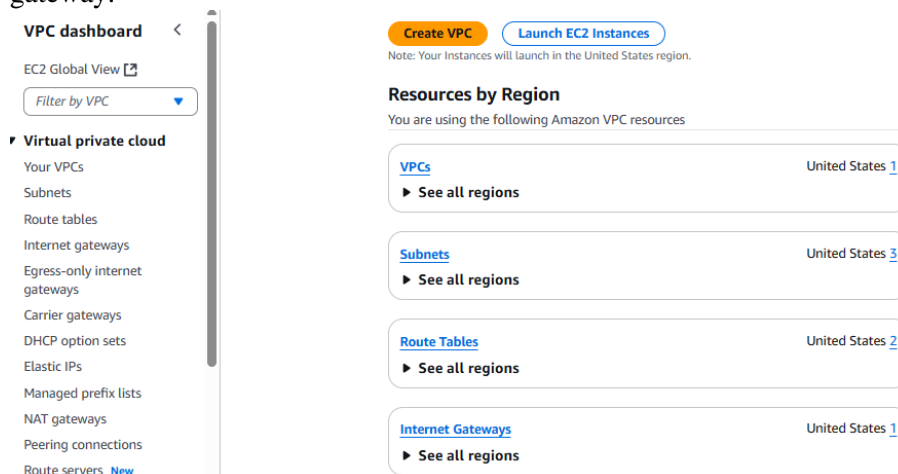
1. Create Web_VPC Subnets and Attach a New Internet Gateway

1.1. Create a VPC

1. Once you are logged in to the AWS Management Console, navigate to **VPC**.



2. On this page, we can see that we already have 1 VPC, 3 subnets, 2 route tables and an internet gateway.

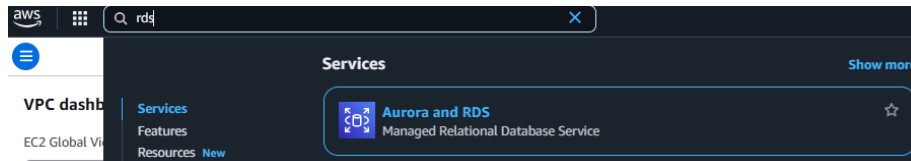


3. Under *Resources by Region*, click **VPCs**.

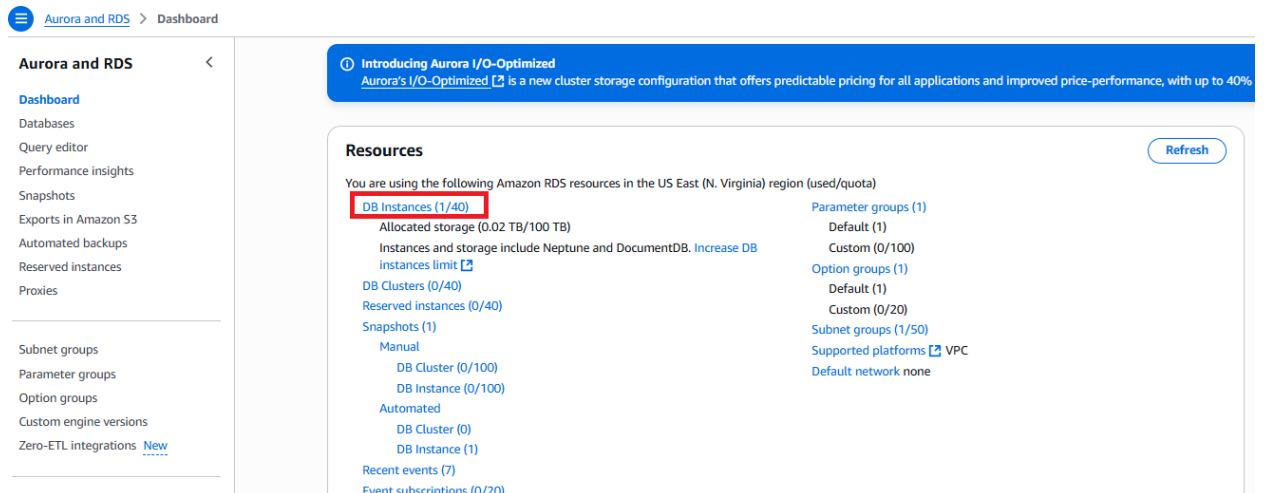
4. As you can see, we have our database VPC that was created with this hands-on lab.



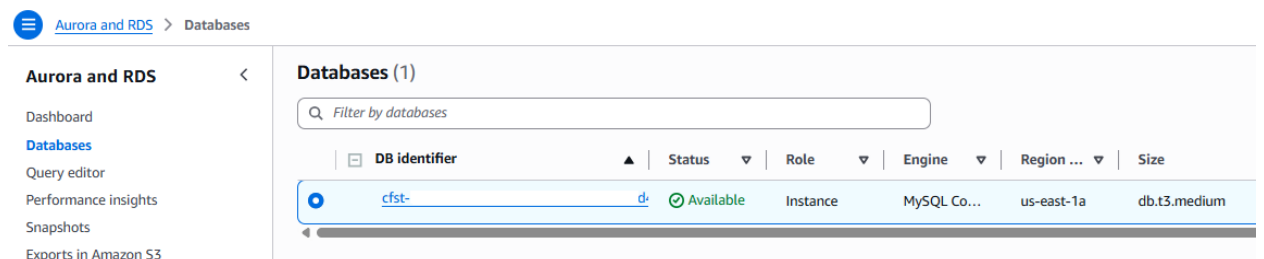
5. Use the top search bar to look for and navigate to **RDS** in a new tab.



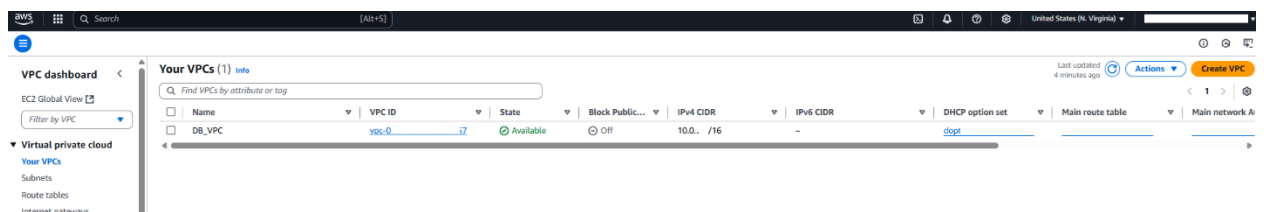
6. Click **DB Instances**, and observe the instance created for this lab.



Note: Keep this tab open for use later in the lab.



7. Go back to your VPC tab and click **Create VPC**.



8. Ensure the **VPC only** option is selected.
9. Set the following values:
 - a. **Name tag**: Enter **Web_VPC**.
 - b. **IPv4 CIDR block**: Enter **192.168.<number>.<number>/16**. The IPv4 CIDR block will need to be different from the database VPC (**10.0.<number>.<number>/16**) that was created with this lab.
10. Leave the rest of the settings as their defaults, and click **Create VPC**.

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

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

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

IPv4 CIDR

CIDR block size must be between /16 and /28.

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](#)

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

1.2. Create a Subnet

1. On the left menu under **VIRTUAL PRIVATE CLOUD**, select **Subnets**.
2. Click **Create subnet**.

VPC dashboard < Subnets (3) [Info](#)

EC2 Global View [Filter by VPC](#)

Virtual private cloud

Your VPCs

Subnets

Route tables

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID	Availi...
Public	subnet-01	5	Available	vpc-	10B...	Off	10. 1/24	250
Private1	subnet-01	19	Available	vpc-	10B...	Off	10. 1/24	250
Private2	subnet-01	12	Available	vpc-	10B...	Off	10. 1/24	251

Last updated 19 minutes ago

[Actions](#) [Create subnet](#)

3. For **VPC ID**, select the newly created **Web_VPC**.

Create subnet [Info](#)

VPC
VPC ID
 Create subnets in this VPC.

Select a VPC

Q

vpc-0	7 (DB_VPC)
vpc-05	fc (Web_VPC)

4. Under **Subnet settings**, set the following values:
 - a. **Subnet name**: Enter **WebPublic**.
 - b. **Availability Zone**: Select **us-east-1a**.
 - c. **IPv4 CIDR block**: Enter **192.168.<number>.<number>/24**.
5. Click **Create subnet**.

Subnet settings
 Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
 Create a tag with a key of "Name" and a value that you specify.

WebPublic

The name can be up to 256 characters long.

Availability Zone [Info](#)
 Choose the zone in which your subnet will reside, or let Amazon choose one for you.

United States (N. Virginia) / us-east-1a

IPv4 VPC CIDR block [Info](#)
 Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

192.168. /16

IPv4 subnet CIDR block

192.168. /24 256 IPs

< > ^ v

▼ **Tags - optional**

Key	Value - optional	
Q Name	Q WebPublic	Remove

[Add new tag](#)

You can add 49 more tags.

[Remove](#)

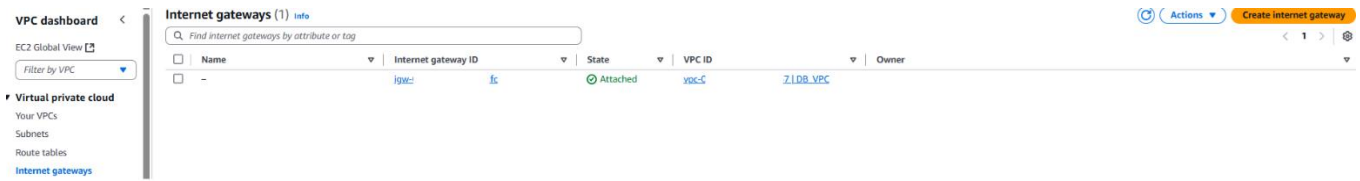
[Add new subnet](#)

[Cancel](#) [Create subnet](#)

Note: For testing purposes, we need our WordPress instance to be available to the internet just to make sure it works, even though our scenario said it's going to be a private internal blog. To do this, we need to set up an internet gateway.

1.3. Create an Internet Gateway

1. On the left menu, select **Internet Gateways**.
2. Click **Create internet gateway**.



3. For *Name tag*, enter **WebIG**.
4. Click **Create internet gateway**.

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

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

WebIG

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**

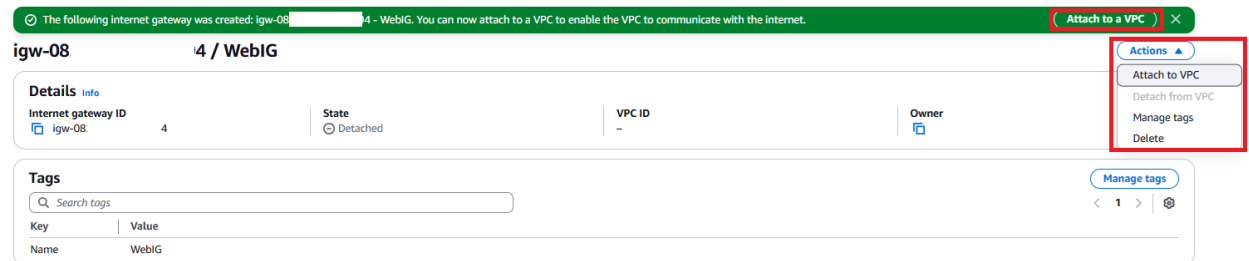
Q Name X Q WebIG X Remove

Add new tag

You can add 49 more tags.

Cancel Create internet gateway

5. In the green notification at the top of the page, click **Attach to a VPC**. Another option is **Actions** → **Attach to a VPC**.



6. In *Available VPCs*, select the **Web_VPC** and click **Attach internet gateway**.

Attach to VPC (igw-08) **94** Info

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

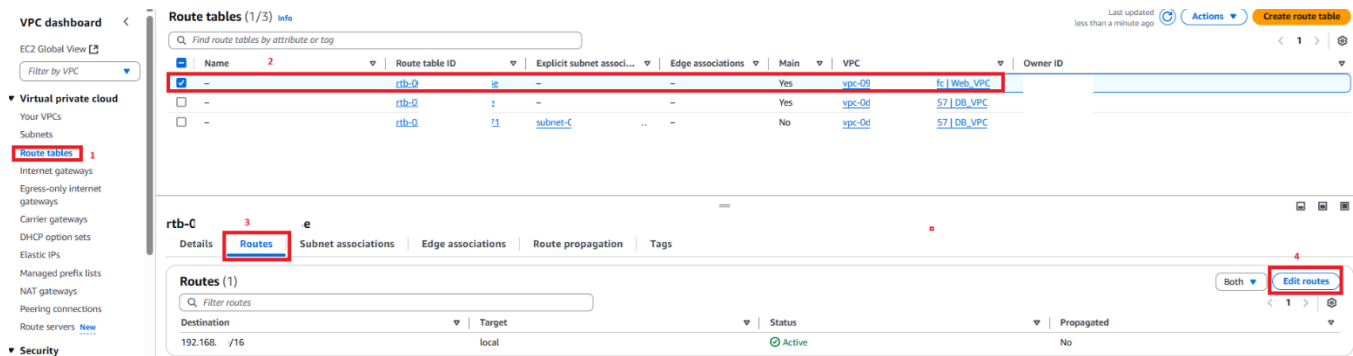
Q Select a VPC

vpc-09 fc - Web_VPC

AWS Command Line Interface Command

Cancel Attach internet gateway

7. On the left menu, select **Route Tables**.
8. Select the checkbox for the **Web_VPC**.
9. Underneath, select the *Routes* tab and click **Edit routes**.



10. Click **Add route**.

11. Set the following values:

- Destination:** Enter **0.0.0.0/0**.
- Target:** Select **Internet Gateway**, and select the internet gateway that appears in the list.

12. Click **Save changes**.

Edit routes

Note: So now, anything in this subnet that tries to get to anything that isn't the 192.168 address, which is our local address, will go out to the internet via this internet gateway. Next, we'll link our 2 VPCs together with the peering connection.

2. Create a Peering Connection

- On the left menu, select **Peering Connections**.
- Click **Create peering connection**.

VPC dashboard < **Peering connections** info

EC2 Global View  Filter by VPC

- Virtual private cloud
 - Your VPCs
 - Subnets
 - Route tables
 - Internet gateways
 - Egress-only internet gateways
 - Carrier gateways
 - DHCP option sets
 - Elastic IPs
 - Managed prefix lists
 - NAT gateways
 - Peering connections** ¹

Peering connections info

Find peering connections by attribute or tag

Name	Peering connection ID	Status	Requester VPC	Accepter VPC	Requester CIDRs	Accepter CIDRs
No peering connection found						

Select a peering connection above

3. Set the following values:

- Name**: Enter **DBtoWeb**.
- VPC (Requester)**: Select the **DB_VPC**.
- VPC (Accepter)**: Select the **Web_VPC**.

4. Click **Create peering connection**.

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. [Info](#)

Peering connection settings

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

DBtoWeb

Select a local VPC to peer with

VPC ID (Requester)
vpc- (DB_VPC)

VPC CIDRs for vpc- (DB_VPC)

CIDR	Status	Status reason
10.0. /16	Associated	-

Select another VPC to peer with

Account
☒ My account
☐ Another account

Region
☒ This Region (us-east-1)
☐ Another Region

VPC ID (Accepter)
vpc- (Web_VPC)

VPC CIDRs for vpc- (Web_VPC)

CIDR	Status	Status reason
192.168. /16	Associated	-

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
 × Remove

Add new tag
You can add 49 more tags.

Cancel Create peering connection

5. This will attempt to link these 2 VPCs together. Here on the details page for our new peering connection, we see that we have a notice that our peering connection is pending acceptance. You can imagine if you were working between different accounts, once you've created a peering

connection, someone on another account would have to accept it. In this case since we control both sides of this VPC peering connection, we can accept the connection ourselves. The status will change to active.

- At the top of the page, click **Actions** → **Accept request**.

pcx-0t **51 / DBtoWeb**

Pending acceptance
You can accept or reject this peering connection request using the 'Actions' menu. You have until Friday, May 9, 2025 at 22:37:42 GMT+3 to accept or reject the request, otherwise it expires.

Details [Info](#)

Requester owner ID
[pcx-0t](#) [i51](#)

Peering connection ID
[pcx-0t](#) [i51](#)

Status
[Pending Acceptance by](#)

Expiration time
Friday, May 9, 2025 at 22:37:42 GMT+3

Accepter owner ID
[vpc-0t](#) [7 / DB_VPC](#)

Requester VPC
[vpc-0t](#) [7 / DB_VPC](#)

Requester CIDRs
[10.0.0.0 /16](#)

Requester Region
[N. Virginia \(us-east-1\)](#)

VPC Peering connection ARN
[arn:aws:ec2:us-east-1::vpc-peering-connection/pcx-0t](#)

Accepter VPC
[vpc-0t](#) [fc / Web_VPC](#)

Accepter CIDRs
-

Accepter Region
[N. Virginia \(us-east-1\)](#)

DNS | **Route tables** | **Tags**

DNS settings [Edit DNS settings](#)

Requester VPC ([vpc-0t](#)) [57 / DB_VPC](#) [Info](#)

Allow accepter VPC to resolve DNS of hosts in requester VPC to private IP addresses
☐ Disabled

Accepter VPC ([vpc-0t](#)) [fc / Web_VPC](#) [Info](#)

Allow requester VPC to resolve DNS of hosts in accepter VPC to private IP addresses
☐ Disabled

- Click **Accept request**.

Accept VPC peering connection request [Info](#)

Are you sure you want to accept this VPC peering connection request? ([pcx-0t](#) [51 / DBtoWeb](#))

Requester VPC
[vpc-0t](#) [57 / DB_VPC](#)

Accepter VPC
[vpc-0t](#) [fc / Web_VPC](#)

Requester CIDRs
[10.0.0.0 /16](#)

Accepter CIDRs
-

Requester Region
[N. Virginia \(us-east-1\)](#)

Accepter Region
[N. Virginia \(us-east-1\)](#)

Requester owner ID
[\(This account\)](#)

Accepter owner ID
[\(This account\)](#)

[Cancel](#) [Accept request](#)

- On the left menu, select **Route Tables**.

- Select the checkbox for the **Web_VPC**.

- Underneath, select the **Routes** tab, and click **Edit routes**.

VPC dashboard < **Route tables (1/3)** [Info](#)

EC2 Global View [Filter by VPC](#)

Virtual private cloud
Your VPCs
Subnets
Route tables [1](#)
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers [New](#)
Security

Route tables (1/3) [Info](#)

[Find route tables by attribute or tag](#)

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
<input checked="" type="checkbox"/>	rtb-0t	rtb-0t	lg	-	Yes	vpc-0t 1 / Web_VPC	
<input type="checkbox"/>	rtb-0t	rtb-0t	z	-	Yes	vpc-0t 7 / DB_VPC	
<input type="checkbox"/>	rtb-0t	rtb-0t	z1	subnet-0t	No	vpc-0t 7 / DB_VPC	

rtb-0t [3](#)

Routes | **Subnet associations** | **Edge associations** | **Route propagation** | **Tags**

Routes (2) [Both](#) [Edit routes](#) [4](#)

[Filter routes](#)

Destination	Target	Status	Propagated
0.0.0.0/0	lg	Active	No
192.168.0.0/16	local	Active	No

11. Click **Add route**.

12. Set the following values:

- Destination:** Enter **10.0.<number>.<number>/16**.
- Target:** Select **Peering Connection**, and select the peering connection that appears in the list.

13. Click **Save changes**.

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	Active	No
10.0.0.0/16			No

Buttons: Add route, Cancel, Preview, Save changes

Target dropdown options: Carrier Gateway, Core Network, Egress Only Internet Gateway, Gateway Load Balancer Endpoint, Instance, Internet Gateway, local, NAT Gateway, Network Interface, Outpost Local Gateway, Peering Connection, Transit Gateway, Virtual Private Gateway

Note: We're going to do the same thing for our database VPC.

14. Go back to **Route Tables**, and select the checkbox for the **DB_VPC** instance with a **Main** column value of **Yes**.

15. Underneath, select the **Routes** tab, and click **Edit routes**.

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
rtb-0	8			Yes	vpc-	7 Web_VPC
rtb-0	8			Yes	vpc-	7 DB_VPC
rtb-0	71	subnet-4		No	vpc-	7 DB_VPC

Buttons: Edit routes

16. Click **Add route**.

17. Set the following values:

- Destination:** Enter **192.168.<number>.<number>/16**.

- b. **Target:** Select Peering Connection, and select the peering connection that appears in the list.

18. Click **Save changes**.

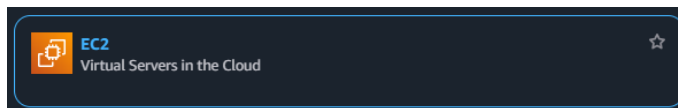
Edit routes

The screenshot shows the 'Edit routes' interface in the AWS Management Console. It features a table with columns for Destination, Target, Status, and Propagated. The first row shows a destination of '10.0 /16' and a target of 'local' with an 'Active' status. A search bar for the target is visible, showing 'Q local'. A dropdown menu is open below the search bar, listing various targets such as 'Carrier Gateway', 'Core Network', 'Egress Only Internet Gateway', 'Gateway Load Balancer Endpoint', 'Instance', 'Internet Gateway', 'local', 'NAT Gateway', 'Network Interface', 'Outpost Local Gateway', 'Peering Connection' (highlighted), 'Transit Gateway', and 'Virtual Private Gateway'. The 'Add route' button is highlighted in the bottom left. The 'Save changes' button is highlighted in the bottom right.

Note: Our VPCs can communicate with each other now.

3. Create an EC2 Instance and Configure WordPress

1. In a new browser tab, navigate to **EC2**.



2. Click **Launch instance** → **Launch instance**.

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.



Note: Your instances will launch in the United States (N. Virginia) Region

3. Give a name to the instance. Scroll down and under **Quick Start**, select the **Ubuntu** image box.
4. Under **Amazon Machine Image (AMI)**, click the dropdown and select **Ubuntu Server 24.04 LTS**.

Launch an instance [Info](#)

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 [Info](#)

Name

MyInstance

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

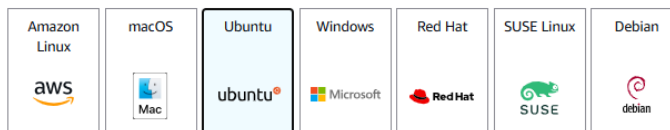
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

Recents

My AMIs

Quick Start



[Browse more AMIs](#)
Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami- (64-bit (x86)) / ami- (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86)

AMI ID

ami-

Publish Date

2025-03-05

Username

ubuntu

Verified provider

5. Under **Instance type**, click the dropdown and select **t3.micro**.

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.micro
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour

[Additional costs apply for AMIs with pre-installed software](#)

6. In this lab scenario, for **Key pair**, click the dropdown and select **Proceed without a key pair**. For production environment you should create a key pair or use an existing key pair.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Proceed without a key pair (Not recommended)

Default value

[Create new key pair](#)

7. In the **Network settings** section, click the **Edit** button.

▼ Network settings [Info](#)

Network

vpc- | DB_VPC

Subnet

subnet- : | Public

Auto-assign public IP

Disable

Edit

- ▼ Network settings

VPC - required | Info

vpc-
192.168 I/16 (Web_VPC)

⌵ ↻

Subnet | Info

subnet-
VPC: vpc- Owner: Availability Zone: us-east-1a WebPublic
Zone type: Availability Zone IP addresses available: 251 CIDR: 192.168 /24

⌵ ↻ Create new subnet

Auto-assign public IP | Info

Enable

⌵

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

- Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Security group name - required

launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./!@#,%&*~:|;{}\$*

Description - required | Info

launch-wizard-1 created 2025-05-02T20:22:47.139Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | Info

ssh

Source type | Info

Anywhere

Protocol | Info

TCP

Port range | Info

22

Source | Info

Add CIDR, prefix list or security group

0.0.0.0/0

Description - optional | Info

e.g. SSH for admin desktop

Remove

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type | Info

HTTP

Source type | Info

Custom

Protocol | Info

TCP

Port range | Info

80

Source | Info

Add CIDR, prefix list or security group

0.0.0.0/0

Description - optional | Info

e.g. SSH for admin desktop

Remove

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.

Add security group rule

12. Scroll to the bottom, and expand **Advanced details**.

► Advanced details [Info](#)

13. At the bottom, under *User data*, copy and paste the following bootstrap script:

```
#!/bin/bash

sudo apt update

sudo apt install apache2 php libapache2-mod-php php-mysql php-curl php-gd php-mbstring
php-xml php-xmlrpc php-soap php-intl php-zip unzip -y

sudo ufw allow in "Apache"

sudo a2enmod rewrite

systemctl restart apache2

cd /tmp/ && wget https://wordpress.org/latest.zip

unzip latest.zip -d /var/www

chown -R www-data:www-data /var/www/wordpress/

mv /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-config.php

cd /var/www/wordpress/

perl -pi -e "s/database_name_here/wordpress/g" wp-config.php
perl -pi -e "s/username_here/wordpress/g" wp-config.php
perl -pi -e "s/password_here/wordpress/g" wp-config.php

perl -i -pe'

BEGIN {

@chars = ("a" .. "z", "A" .. "Z", 0 .. 9);

push @chars, split //, "!@#\$%^&*()-_[]{}<>~\`+=,.;:/?|";

sub salt { join "", map $chars[ rand @chars ], 1 .. 64 }

}

s/put your unique phrase here/salt()/ge

' wp-config.php

wget https://raw.githubusercontent.com/ACloudGuru-Resources/course-aws-certified-
solutions-architect-associate/main/lab/5/000-default.conf

mkdir wp-content/uploads

chmod 775 wp-content/uploads
```

mv 000-default.conf /etc/apache2/sites-enabled/

systemctl restart apache2

14. At the bottom, click **Launch Instance**.

Note: It may take a few minutes for the new instance to launch. The script is going to install our Apache web server and get WordPress installed when the instance is created, and then we can go in and make a few changes to allow our WordPress installation to work properly. We will need to point our WordPress instance to the RDS database that was created with this lab environment.

2

Allow tags in metadata [Info](#)

Select

User data - optional [Info](#)
Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/bin/bash
sudo apt update
sudo apt install apache2 php libapache2-mod-php php-mysql php-curl php-gd php-mbstring php-xml
php-xmlrpc php-soap php-intl php-zip unzip -y
sudo ufw allow in "Apache"
sudo a2enmod rewrite
systemctl restart apache2
cd /tmp/ && wget https://wordpress.org/latest.zip
unzip latest.zip -d /var/www
chown -R www-data:www-data /var/www/wordpress/
mv /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-config.php
cd /var/www/wordpress/
perl -pi -e "s/database_name_here/wordpress/g" wp-config.php
perl -pi -e "s/username_here/wordpress/g" wp-config.php
perl -pi -e "s/password_here/wordpress/g" wp-config.php
perl -pi -pe 'BEGIN {
@chars = ("a".."z", "A".."Z", "0".."9");
push @chars, split //, " !@#\$%^&*()-_ []{}<>~\`+~,:;/?";
sub salt { join "", map { $chars[ rand @chars ], 1 .. 64 } }
}
s/put your unique phrase here/salt()/ge
' wp-config.php
wget https://raw.githubusercontent.com/ACloudGuru-Resources/course-aws-certified-solutions-
architect-associate/main/lab/5/000-default.conf
mkdir wp-content/uploads
chmod 775 wp-content/uploads
mv 000-default.conf /etc/apache2/sites-enabled/
systemctl restart apache2
```

☐ User data has already been base64 encoded

Summary

Number of instances [Info](#)

1

Software image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-084568db458526404

Virtual server type (instance type)
t3.micro

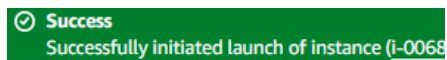
Firewall (security group)
New security group

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 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Preview code](#)

15. From the green box that appears after the instance launches, open the link for the instance in a new browser tab.



16. Observe the **Instance state** column, and check to ensure it is **Running** before you proceed.

17. Select the checkbox for the new instance and click **Connect**.

Instances (1/1) [Info](#)

Find Instance by attribute or tag (case-sensitive) All states

Instance ID: i-0066... [Clear filters](#)

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Moni
<input checked="" type="checkbox"/>	MyInstance	i-0066...	Running View logs	t3.micro	Initializing	View alarms	us-east-1a	-	-	-	-	disab

18. Click **Connect**.

Note: The startup script for the instance may take a few minutes to complete and you may need to wait for it to complete before proceeding with the next step.

Connect to instance [info](#)

Connect to your instance i-0068l (MyInstance) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-0068l (MyInstance)

Connection Type

☒ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

☒ Public IPv4 address
☐ IPv6 address

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

Q ubuntu X

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel Connect

19. To confirm WordPress installed correctly, view the configuration files in the CLI:

cd /var/www/wordpress

ls

```

ubuntu@ip-192-168-1-254:~$ cd /var/www/wordpress
ubuntu@ip-192-168-1-254:~$ ls
index.php  readme.html  wp-admin  wp-comments-post.php  wp-content  wp-includes  wp-load.php  wp-mail.php  wp-signup.php  xmlrpc.php
license.txt  wp-activate.php  wp-blog-header.php  wp-config.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
ubuntu@ip-192-168-1-254:~$

```

Note: Now we need to configure WordPress and we're going to edit a config file.

20. To configure WordPress, open **wp-config.php**: **sudo vim wp-config.php**

```

sudo vim wp-config.php
**
* The base configuration for WordPress
*
* The wp-config.php creation script uses this file during the installation.
* You don't have to use the website, you can copy this file to "wp-config.php"
* and fill in the values.
*
* This file contains the following configurations:
*
* * Database settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
*
* @package WordPress
*/

/** Database settings - You can get this info from your web host ** */
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wordpress' );

/** Database password */
define( 'DB_PASSWORD', 'wordpress' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

```

21. Go back to your browser tab with RDS.

22. Click the link to open the provisioned RDS instance.

23. Under **Connectivity & security**, copy the RDS **Endpoint**.

Aurora and RDS > Databases > cfst- -database-8i

Aurora and RDS

- Dashboard
- Databases**
- Query editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies

Subnet groups

Parameter groups

Option groups

Custom engine versions

Zero-ETL integrations [New](#)

cfst-3372-4 **-database-**

Summary

DB identifier
cfst-3372-

Status
Available

Class
db.t3.medium

Role
Instance

Current activity
0 Connection

CPU
3.60%

Connectivity & security | Monitoring | Logs & events | Configuration | Zero-ETL integrations | Ma

Connectivity & security

Endpoint & port

Endpoint
cfst-3372-
-database-
.us-east-1.rds.amazonaws.com

Networking

Availability Zone
us-east-1a

24. Go back to the tab with the terminal, and scroll down to **/** Database hostname */**.
25. Press “i” to enter Insert mode.
26. Replace **localhost** with the RDS endpoint you just copied. Ensure it remains wrapped in single quotes.
27. Press “**ESC**” followed by “:**wq**”, and press **Enter**. Leave this tab open.

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );


/** Database username */
define( 'DB_USER', 'wordpress' );

/** Database password */
define( 'DB_PASSWORD', 'wordpress' );

/** Database hostname */
define( 'DB_HOST', 'cfst-3372- -database- .us-east-1.rds.amazonaws.com' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```



4. Modify the RDS Security Groups to Allow Connections from the Web_VPC VPC

1. Go back to your RDS browser tab.
2. In **Connectivity & security**, click the active link under **VPC security groups**.

The screenshot shows the Amazon RDS console for a database instance named 'cfst-3372'. The 'Connectivity & security' tab is selected. The 'VPC security groups' section is highlighted with a red box, showing the 'DatabaseSecurityGroup-b' security group. The 'Endpoint & port' section shows the endpoint 'cfst-3372' and port '3306'. The 'Networking' section shows the VPC 'DB_VPC (vpc-...)' and subnet group 'cfst-3372-...'. The 'Security' section shows the security group 'DatabaseSecurityGroup-b' with a status of 'Active'.

3. Checkmark the **DatabaseSG** Security Group.
4. At the bottom, select the **Inbound rules** tab.
5. Click **Edit inbound rules**.

The screenshot shows the Amazon RDS console for the 'DatabaseSG' security group. The 'Inbound rules' tab is selected. The 'Edit inbound rules' button is highlighted with a red box. The 'Inbound rules' table shows one rule with the following details:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sg-	IPv4	MySQL/Aurora	TCP	3306	10.0./16	-

6. Click **Add rule**.
7. Under **Type**, search for and select **MYSQL/Aurora**.
8. Under **Source**, search for and select **192.168.<number>.<number>/16**.
9. Click **Save rules**.

Edit inbound rules [Info](#)

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

Inbound rules [Info](#)

Security group rule ID

sg-

Type [Info](#)

MySQL/Aurora

Protocol [Info](#)

TCP

Port range [Info](#)

3306

Source [Info](#)

Custom

Description - optional [Info](#)

10.0 /16 X

Delete

-

MySQL/Aurora

TCP

3306

Custom

Q 192.168. /16 X

192.168. /16 X

Delete

1

Add rule

2

3

4

Cancel

Preview changes

Save rules

Note: These 2 rules will allow anyone from the **10.0**. or the **192.168**. VPCs to connect to the database.

10. Return to the terminal page.

11. Below the terminal window, copy the public IP address of your server.


i-0068 **(MyInstance)**
PublicIPs: 5 2 PrivateIPs: *

12. Open a new browser tab and paste the public IP address in the address bar. You should now see the WordPress installation page.

13. Set the following values:

- Site Title:** Enter **A Blog Guru**.
- Username:** Enter **guru**.
- Your Email:** Enter **test@test.com**.

14. Click **Install WordPress**.



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

A Blog Guru

Username

guru

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

.....

Strong

Show

Important: You will need this password to log in. Please store it in a secure location.

Your Email

test@test.com

Double-check your email address before continuing.

Search engine visibility

☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

15. Reload the public IP address in the address bar to view your newly created WordPress blog.



In this lab, we had an existing database VPC, and we created a new website VPC. We created a subnet, we edited our route table, and then we peered our 2 VPCs together so they could talk to each other. We also edited our route tables to ensure that traffic was flowing in the correct directions. We created our EC2 instance in the web VPC, and we configured WordPress to connect to our RDS instance, which exists in our database VPC.