



UNIVERSIDADE
LUSÓFONA

Sistemas de Informação na Nuvem

Relatório Laboratório 4

Tomás Nave, a22208623

André Jesus, a22207061

Índice

Índice	2
Introdução.....	4
Exercício 1 – Criar infraestrutura presente na figura 1	5
Criação da VPC Pública e Subnets.....	5
Criação da VPC Privada e Subnets	6
Route Tables Publica	7
Route Tables Privada	7
Peering Connection	8
SG1 – Bastion.....	8
SG2 – WordPress.....	9
SG3 – DB.....	10
Exercício 2 – Criar as instâncias EC2	11
Vm – Bastion	11
Vm – WordPress	12
Exercício 3 – Criar Serviço RDS	13
Exercício 4 – Configuração das Instâncias	14
3.1 Configuração Instância DB:	Erro! Marcador não definido.
3.1.1 Acesso á Instância DB a partir do Bastion.....	Erro! Marcador não definido.
3.1.2 Instalação Maria DB	Erro! Marcador não definido.
3.1.3 Criação da Base de Dados e Utilizador	Erro! Marcador não definido.
3.2 Configuração Instância WordPress:	14
3.2.1 Acesso á Instância Wordpress a partir do Bastion	14
3.2.2 Instalação nginx	14
3.2.3 Instalação php	15
3.2.3 Instalação WordPress.....	16
3.2.4 Transferência dos Ficheiros do WordPress para nginx.....	16
3.2.5 Configuração do wp-config.php	18
3.2.6 Ajuste de permissões e limpeza de ficheiros padrão	19
3.2.7 Configuração do Nginx para interpretar PHP	19
3.2.8 Configuração do ficheiro default.conf	20
3.2.9 Teste Final e Instalação WordPress via Browser.....	20

Conclusão..... 22

Introdução

O principal objetivo deste trabalho foi criar uma infraestrutura completa e segura na plataforma AWS (Amazon Web Services), composta por máquinas virtuais (instâncias EC2), uma base de dados relacional gerida (RDS), e duas redes virtuais separadas (VPCs), ligadas entre si através de VPC Peering. Esta infraestrutura simula um ambiente real de alojamento de uma aplicação web baseada em WordPress, com uma base de dados externa, garantindo o isolamento, controlo de acesso e segmentação da rede.

Cada VPC foi estruturada com subnets configuradas consoante o seu papel. A VPC pública aloja duas instâncias EC2: uma Bastion Host, que serve de ponto de acesso seguro por SSH, e a instância WordPress, acessível pela Internet via HTTP/HTTPS. A segunda VPC, embora chamada "VPC-Private" por consistência com o diagrama fornecido no enunciado, contém na realidade **subnets públicas**, uma vez que estas estão associadas a uma Internet Gateway e permitem a ligação à Internet — necessária para que os serviços nelas alojados consigam aceder ao exterior.

A instância de base de dados (Amazon RDS MySQL) foi colocada nesta VPC e, apesar de não ser publicamente acessível (isto é, não possui endereço IP público), precisa de aceder a serviços externos (como atualizações ou resolução de nomes DNS). Para permitir este acesso de saída à Internet de forma segura, foi configurada uma **NAT Gateway** numa das subnets públicas da VPC, que permite às instâncias privadas realizarem comunicação de saída sem estarem diretamente expostas à Internet.

A comunicação entre a instância WordPress (na VPC pública) e a base de dados (na VPC-Private) é feita através do **endpoint DNS privado gerado pela RDS**, que é resolvido internamente via peering. No entanto, como o RDS não tem um IP público nem é acessível diretamente, o WordPress precisa, por vezes, de resolver este domínio ou comunicar com o DNS da AWS — o que justifica o uso da NAT Gateway. Esta abordagem cumpre os princípios de segurança e garante que a base de dados continua inacessível diretamente do exterior, mantendo o isolamento pretendido.

Para garantir a segurança e o correto funcionamento da comunicação entre os componentes, foram definidas regras específicas nas ACLs (Listas de Controlo de Acesso) e nos Security Groups (Grupos de Segurança). Cada um dos grupos de segurança foi configurado para permitir apenas o tráfego essencial para o funcionamento da aplicação, como conexões SSH, HTTP, HTTPS e acesso à base de dados na porta 3306. As regras de outbound foram abertas para todos os destinos, garantindo a comunicação de saída.

A criação do serviço RDS exigiu a definição de um grupo de subnets em diferentes zonas de disponibilidade (AZs), tal como exigido pela AWS, para garantir tolerância a falhas. O

acesso à base de dados foi testado a partir da instância WordPress, simulando um cenário de produção em que o acesso é feito de forma controlada e indireta.

No final do processo, foram realizados testes de conectividade entre as várias instâncias e validada a ligação do WordPress ao serviço RDS, garantindo que a aplicação consegue aceder e utilizar a base de dados externa. Esta infraestrutura demonstra a importância da

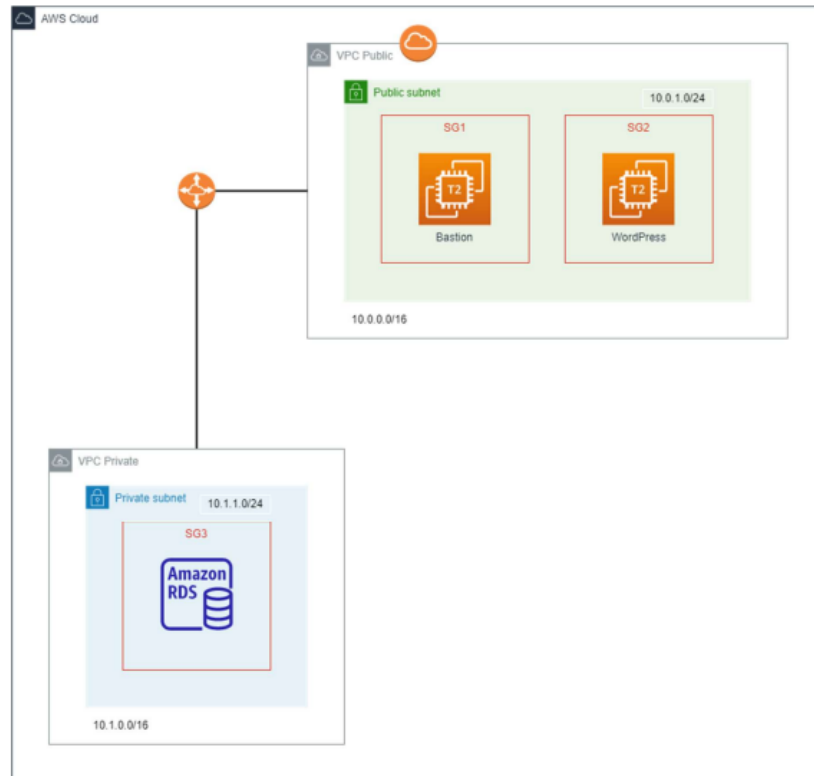


Figura 1 - Infraestrutura Objetivo

Exercício 1 – Criar infraestrutura presente na figura 1

Criação da VPC Pública e Subnets

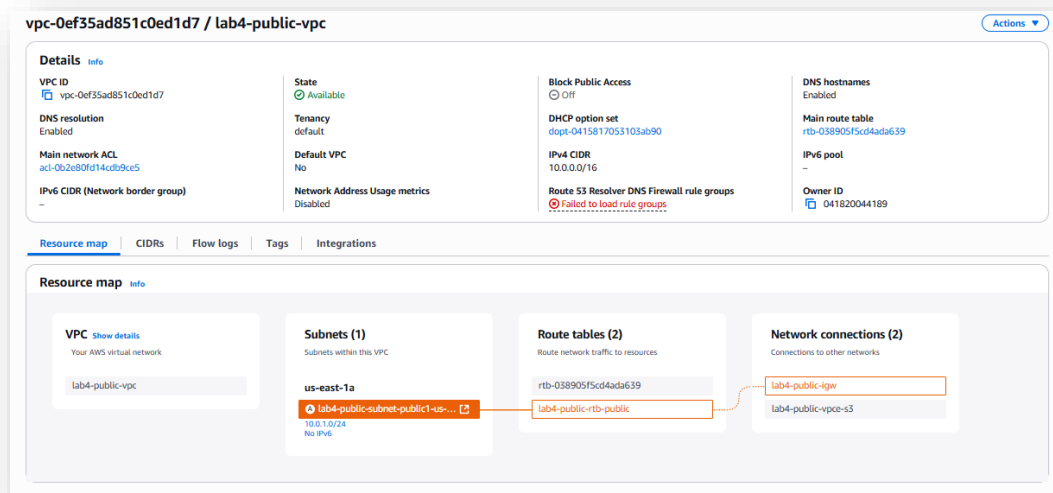


Figura 2 - VPC Publica e Subnet

Criação da VPC Privada e Subnets

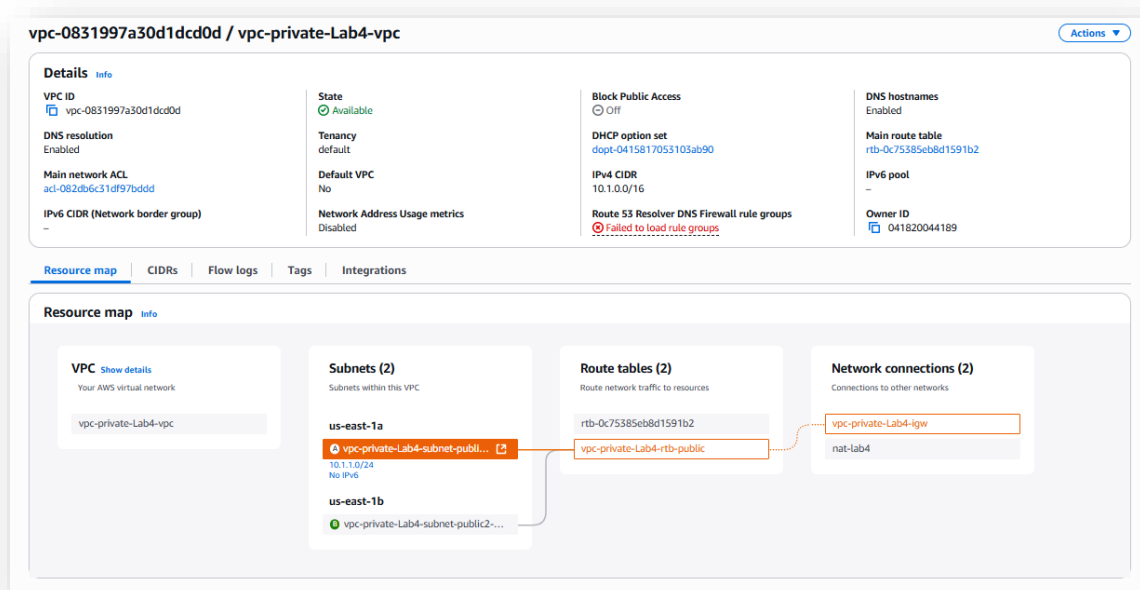


Figura 3 - VPC Privada e Subnet

Route Tables Publica

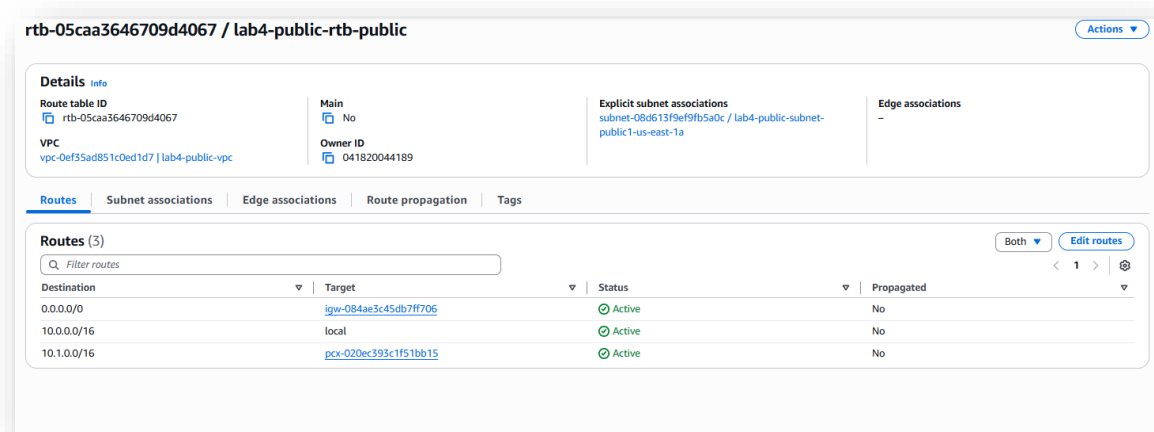


Figura 4 - Route Table Publica

Route Tables Privada

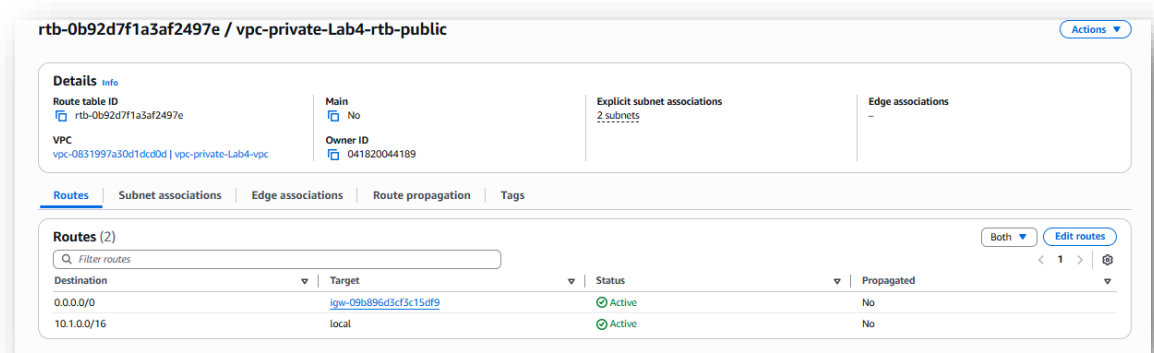


Figura 5 - Route Table Privada

Peering Connection

pcx-0f2be44f3ba11cb8c / Lab4-peering

Actions

Details

Requester owner ID

041820044189

Peering connection ID

pcx-0f2be44f3ba11cb8c

Status

Active

Expiration time

-

Accepter owner ID

041820044189

Requester VPC

vpc-0831997a30d1dcd0d / vpc-private-Lab4-vpc

Requester CIDRs

10.1.0.0/16

Requester Region

N. Virginia (us-east-1)

VPC Peering connection ARN

arn:aws:ec2:us-east-1:041820044189:vpc-peering-connection/pcx-0f2be44f3ba11cb8c

Accepter VPC

vpc-0ef35ad851c0ed1d7 / lab4-public-vpc

Accepter CIDRs

10.0.0.0/16

Accepter Region

N. Virginia (us-east-1)

DNS

Route tables

Tags

DNS settings

Edit DNS settings

Requester VPC

vpc-0831997a30d1dcd0d / vpc-private-Lab4-vpc

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

Disabled

Accepter VPC

vpc-0ef35ad851c0ed1d7 / lab4-public-vpc

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

Disabled

Figura 6 - Peering Connection

SG1 – Bastion

sg-07be30b5e3e955246 - sg1-bastion-lab4

Actions

Details

Security group name

sg1-bastion-lab4

Owner

041820044189

Security group ID

sg-07be30b5e3e955246

Inbound rules count

1 Permission entry

Description

.

Outbound rules count

1 Permission entry

VPC ID

vpc-0ef35ad851c0ed1d7

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (1)

Manage tags

Edit inbound rules

Search

< 1 >

Name

Security group rule ID

IP version

Type

Protocol

Port range

Source

Description

-

sg-013702ff5862649a8

IPv4

SSH

TCP

22

0.0.0.0/0

-

Figura 7 - SG1 - Bastion Inbound Rules

sg-07be30b5e3e955246 - sg1-bastion-lab4

Details

Security group name
sg1-bastion-lab4

Owner
041820044189

Security group ID
sg-07be30b5e3e955246

Inbound rules count
1 Permission entry

Description
.

Outbound rules count
1 Permission entry

VPC ID
vpc-0ef35ad851c0ed1d7

Inbound rules | **Outbound rules** | Sharing - new | VPC associations - new | Tags

Outbound rules (1)

Search

Manage tags

Edit outbound rules

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
<input type="checkbox"/>	-	sgr-00cad4d9b96a5e9ab	IPv4	All traffic	All	All	0.0.0.0/0	-

Figura 8 - SG1 - Bastion Outbound Rules

SG2 – WordPress

sg-0b8d9545821550de2 - sg2-wordpress-lab4

Details

Security group name
sg2-wordpress-lab4

Owner
041820044189

Security group ID
sg-0b8d9545821550de2

Inbound rules count
3 Permission entries

Description
.

Outbound rules count
1 Permission entry

VPC ID
vpc-0ef35ad851c0ed1d7

Inbound rules | Outbound rules | Sharing - new | VPC associations - new | Tags

Inbound rules (3)

Search

Manage tags

Edit inbound rules

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-0f8e60730da3672e1	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-06a6643eb5e887121	IPv4	HTTP	TCP	80	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0441b75734cd100f5	IPv4	SSH	TCP	22	10.0.1.0/24	-

Figura 9 - SG2 - WordPress Inbound Rules

sg-0b8d9545821550de2 - sg2-wordpress-lab4

Details

Security group name
sg2-wordpress-lab4

Owner
041820044189

Security group ID
sg-0b8d9545821550de2

Inbound rules count
3 Permission entries

Description
.

Outbound rules count
1 Permission entry

VPC ID
vpc-0ef35ad851c0ed1d7

Inbound rules | **Outbound rules** | Sharing - new | VPC associations - new | Tags

Outbound rules (1)

Search

Manage tags

Edit outbound rules

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
<input type="checkbox"/>	-	sgr-0044a8b4b928963c8	IPv4	All traffic	All	All	0.0.0.0/0	-

Figura 10 - SG2 - WordPress Outbound Rules

SG3 – DB

sg-008ed96ecc30acc6c - sgbd-Lab4

Details

Security group name
sgbd-Lab4

Owner
041820044189

Security group ID
sg-008ed96ecc30acc6c

Inbound rules count
2 Permission entries

Description
Created by RDS management console

Outbound rules count
1 Permission entry

VPC ID
vpc-0831997a30d1dcd0d

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (2)

Search

	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sg-029d6bc28c53bcff8	IPv4	MySQL/Aurora	TCP	3306	34.205.28.222/32	-
<input type="checkbox"/>	-	sg-05b38b8f007737077	IPv4	MySQL/Aurora	TCP	3306	62.28.125.114/32	-

Manage tags

Edit inbound rules

Figura 11 - SG3 - DBInstance Inbound Rules

sg-085a4fe0493416c7f - sg3-bd-lab4

Details

Security group name
sg3-bd-lab4

Owner
041820044189

Security group ID
sg-085a4fe0493416c7f

Inbound rules count
1 Permission entry

Description
-

Outbound rules count
1 Permission entry

VPC ID
vpc-01da8e386aeb2d6e5

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Outbound rules (1)

Search

	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
<input type="checkbox"/>	-	sg-0d4616ffd26b0caf3	IPv4	All traffic	All	All	0.0.0.0/0	-

Manage tags

Edit outbound rules

Figura 12 - SG3 - DBInstance Outbound Rules

NAT Gateway

nat-03a5b6d370413939f / nat-lab4

Details

NAT gateway ID
nat-03a5b6d370413939f

NAT gateway ARN
arn:aws:ec2:us-east-1:041820044189:natgateway/nat-03a5b6d370413939f

VPC
vpc-0831997a30d1dcd0d / vpc-private-Lab4-vpc

Connectivity type
Public

Primary public IPv4 address
35.153.154.93

Subnet
subnet-0f85f500c8b625fd4 / vpc-private-Lab4-subnet-public1-us-east-1a

State
Available

Primary private IPv4 address
10.1.1.155

Created
Friday, May 30, 2025 at 10:23:05 GMT+1

State message
-

Primary network interface ID
eni-0c66d0d6cabc151a0

Deleted
-

Secondary IPv4 addresses

Monitoring

Tags

Secondary IPv4 addresses

Search

Private IPv4 address	Network interface ID	Status	Failure message
----------------------	----------------------	--------	-----------------

Edit secondary IPv4 address associations

Secondary IPv4 addresses are not available for this nat gateway.

Figura 13 - NAT Gateway

Exercício 2 – Criar as instâncias EC2

Vm – Bastion

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

vm-bastion-lab4

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

Quick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-0953476d60561c955 (64-bit (x86), uefi-preferred) / ami-05a3e0187917e3e24 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250512.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username
64-bit (x86)	uefi-preferred	ami-0953476d60561c955	2025-05-09	ec2-user

Verified provider

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

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

☒ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ 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*

lab4Key

Create new key pair

▼ Network settings Info

VPC - *required* | Info

vpc-0ef35ad851c0ed1d7 (lab4-public-vpc)

10.0.0.0/16

Subnet | Info

subnet-08d613f9ef9fb5a0c lab4-public-subnet-public1-us-east-1a

VPC: vpc-0ef35ad851c0ed1d7 Owner: 041820044189 Availability Zone: us-east-1a

Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.1.0/24

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

Common security groups | Info

Select security groups

sg1-bastion-lab4 - sg-07be30b5e3e955246

VPC: vpc-0ef35ad851c0ed1d7

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

Figura 14 - Vm Bastion

Vm – WordPress

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

vm-wordpress-lab4

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

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Debian

debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-0953476d60561c955 (64-bit (x86), uefi-preferred) / ami-05a3e0187917e3e24 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250512.0 x86_64 HVM kernel-6.1

Architecture

64-bit (x86)

Boot mode

uefi-preferred

AMI ID

ami-0953476d60561c955

Publish Date

2025-05-09

Username

ec2-user

Verified provider

▼ Instance type

Info

Get advice

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

Free tier eligible

On-Demand Windows base pricing: 0.0162 USD per Hour On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour On-Demand RHEL base pricing: 0.026 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ 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

lab4Key

Create new key pair

▼ Network settings

Info

VPC - required

Info

vpc-0ef35ad851c0ed1d7 (lab4-public-vpc)

10.0.0.0/16

Subnet

Info

subnet-08d613f9ef9fb5a0c lab4-public-subnet-public1-us-east-1a

VPC: vpc-0ef35ad851c0ed1d7 Owner: 041820044189 Availability Zone: us-east-1a

Zone type: Availability Zone IP addresses available: 250 CIDR: 10.0.1.0/24

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

Common security groups

Info

Select security groups

sg2-wordpress-lab4 sg-0b8d9545821550de2

VPC: vpc-0ef35ad851c0ed1d7

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► Advanced network configuration

Figura 15 - Vm - WordPress

Exercício 3 – Criar Serviço RDS

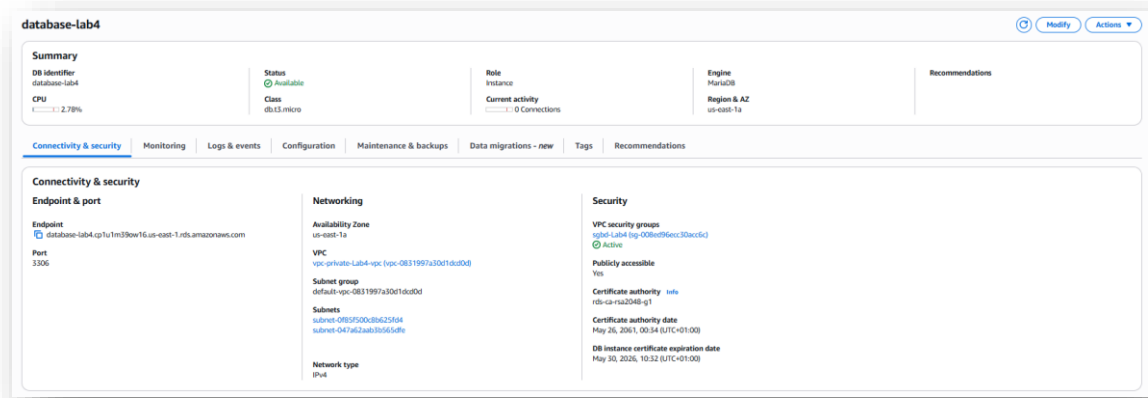


Figura 16 - Serviço RDS

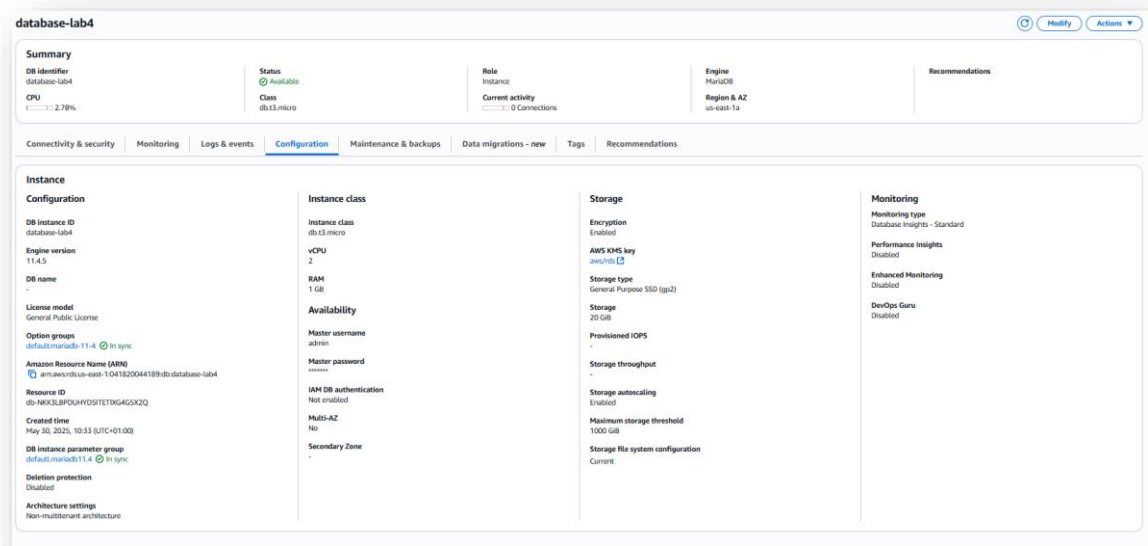


Figura 17 - Serviço RDS configurações

Figura 19 - Instalação nginx

```

Installed:
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  libunwind-1.4.0-5.amzn2023.0.2.x86_64
  nginx-core-1:1.26.3-1.amzn2023.0.1.x86_64
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
  gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
  nginx-1:1.26.3-1.amzn2023.0.1.x86_64
  nginx-filesystem-1:1.26.3-1.amzn2023.0.1.noarch

Complete!
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-05-17 15:49:52 UTC; 16s ago
     Main PID: 26017 (nginx)
       Tasks: 2 (limit: 1111)
      Memory: 2.5M
         CPU: 44ms
    CGroup: /system.slice/nginx.service
            └─26017 "nginx: master process /usr/sbin/nginx"
               └─26018 "nginx: worker process"

May 17 15:49:52 ip-10-0-1-158.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
May 17 15:49:52 ip-10-0-1-158.ec2.internal nginx[26015]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
May 17 15:49:52 ip-10-0-1-158.ec2.internal nginx[26015]: nginx: configuration file /etc/nginx/nginx.conf test is successful
May 17 15:49:52 ip-10-0-1-158.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-0-1-158 ~]$ sudo yum install -y php

```

Figura 20 - Inicialização nginx

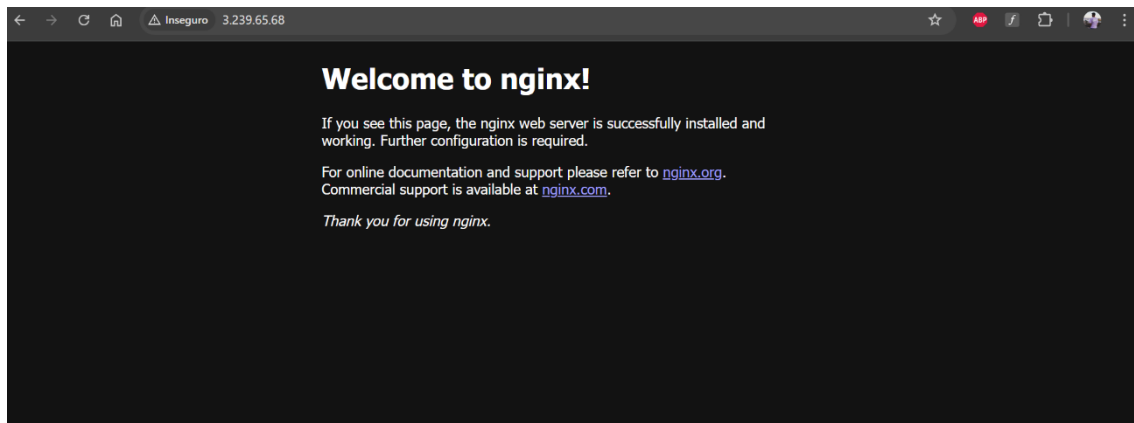


Figura 21 - Acessar a página principal local do servidor para verificar a instalação do nginx

3.1.3 Instalação php

- `sudo yum install -y php`
- `sudo yum install -y php-mysqldb`
- `sudo yum install -y php-fpm`
- `sudo yum install -y php-json`
- `sudo yum install -y php-xml`
- `sudo yum install -y php-mbstring`
- `sudo yum install -y php-gd`
- `sudo yum install -y php-intl`

- php -v / verificar a versão
- sudo systemctl start php-fpm / Inicia o serviço PHP-FPM
- sudo systemctl enable php-fpm / Ativa o PHP-FPM

3.2.3 Instalação WordPress

```
[ec2-user@ip-10-0-1-158 ~]$ php -v
PHP 8.4.6 (cli) (built: Apr  8 2025 19:55:31) (NTS gcc x86_64)
Copyright (c) The PHP Group
Built by Amazon Linux
Zend Engine v4.4.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.4.6, Copyright (c), by Zend Technologies
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl start php-fpm
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl enable php-fpm
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/systemd/system/php-fpm.service.
[ec2-user@ip-10-0-1-158 ~]$ wget https://wordpress.org/latest.tar.gz
--2025-05-17 15:55:24-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)[198.143.164.252]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26926501 (26M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz          100%[=====] 25.68M  38.9MB/s   in 0.7s

2025-05-17 15:55:25 (38.9 MB/s) - 'latest.tar.gz' saved [26926501/26926501]

[ec2-user@ip-10-0-1-158 ~]$ tar -xzf latest.tar.gz
[ec2-user@ip-10-0-1-158 ~]$ sudo rsync -av wordpress/* /usr/share/nginx/html/
```

3.2.4 Transferência dos Ficheiros do WordPress para nginx

1) sudo rsync -av wordpress/* /usr/share/nginx/html/

Este comando copia todos os ficheiros do diretório wordpress para o diretório /usr/share/nginx/html/, que é o local padrão onde o Nginx procura por páginas HTML ou PHP

2) sudo chown -R nginx:nginx /usr/share/nginx/html/

Este comando atribui a propriedade de todos os ficheiros no diretório raiz do site ao utilizador e grupo nginx.

3) sudo nano /etc/nginx/conf.d/wordpress.conf


```

GNU nano 8.3 /etc/nginx/conf.d/wordpress.conf
server {
    listen 80;
    server_name localhost;
    root /usr/share/nginx/html;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ /\.php$ {
        include fastcgi_params;
        fastcgi_pass unix:/var/run/php-fpm/php-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi.conf;
    }

    location ~ /\.ht {
        deny all;
    }
}

```

Figura 22 - Configuração do Nginx para WordPress

3.2.5 Acesso ao RDS através do endpoint

```

[ec2-user@ip-10-0-1-91 ~]$ sudo dnf install mariadb105-client
Last metadata expiration check: 0:44:53 ago on Fri May 30 10:01:56 2025.
No match for argument: mariadb105-client
Error: Unable to find a match: mariadb105-client
[ec2-user@ip-10-0-1-91 ~]$ sudo dnf install mariadb105-server
Last metadata expiration check: 0:45:03 ago on Fri May 30 10:01:56 2025.
Dependencies resolved.
=====
Package                                Architecture Version                                Repository                               Size
=====
Installing:
mariadb105-server                      x86_64      3:10.5.25-1.amzn2023.0.1              amazonlinux                             11 M
Installing dependencies:
mariadb-connector-c                    x86_64      3.3.10-1.amzn2023.0.1                  amazonlinux                             211 k
mariadb-connector-c-config              x86_64      3:10.5.25-1.amzn2023.0.1              amazonlinux                             9.9 k
mariadb105                             x86_64      3:10.5.25-1.amzn2023.0.1              amazonlinux                             1.6 M
mariadb105-common                       x86_64      3:10.5.25-1.amzn2023.0.1              amazonlinux                             29 k
mariadb105-errmsg                       x86_64      3:10.5.25-1.amzn2023.0.1              amazonlinux                             213 k
mysql-selinux                           noarch      1.0.4-2.amzn2023.0.3                  amazonlinux                             36 k
perl-B                                  x86_64      1.80-477.amzn2023.0.6                  amazonlinux                             179 k
perl-DBD-MariaDB                       x86_64      1.22-1.amzn2023.0.4                    amazonlinux                             153 k
perl-DBI                                x86_64      1.643-7.amzn2023.0.3                   amazonlinux                             700 k
perl-Data-Dumper                       x86_64      2.174-460.amzn2023.0.2                 amazonlinux                             55 k
=====

```

Figura 23 - instalação mariadb

```

[ec2-user@ip-10-0-1-91 ~]$ mariadb -h database-lab4.cplu1m39ow16.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 118
Server version: 11.4.5-MariaDB-log managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> select user,host FROM mysql.user;
+-----+-----+
| User      | Host      |
+-----+-----+
| admin     | %         |
| mariadb.sys | localhost |
| rdsadmin  | localhost |
+-----+-----+

```

Figura 24 - Acesso a serviço RDS através do endpoint

```
MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.008 sec)

MariaDB [(none)]> exit
Bye
```

Figura 25 - Criação Base de Dados

3.2.6 Configuração do wp-config.php

```
[ec2-user@ip-10-0-1-158 ~]$ sudo nano /usr/share/nginx/html/wp-config.php/
[ec2-user@ip-10-0-1-158 ~]$ sudo cp /usr/share/nginx/html/wp-config-sample.php /usr/share/nginx/html/wp-config.php
[ec2-user@ip-10-0-1-158 ~]$ sudo nano /usr/share/nginx/html/wp-config.php/
```

1) **sudo nano /usr/share/nginx/html/wp-config.php**

Tentativa de editar o ficheiro de configuração do WordPress. Como não estava criado, criou vazio.

2) **sudo cp /usr/share/nginx/html/wp-config-sample.php /usr/share/nginx/html/wp-config.php**

Copia o ficheiro de exemplo (wp-config-sample.php) para criar o ficheiro real wp-config.php.

3) **sudo nano /usr/share/nginx/html/wp-config.php**

Abre o ficheiro wp-config.php no editor nano para configurar a ligação à base de dados.

```
// ** 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', 'admin' );

/** Database password */
define( 'DB_PASSWORD', 'adminlab4' );

/** Database hostname */
define( 'DB_HOST', 'database-lab4.cp1u1m39ow16.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', '' );
```

Figura 26 - Configuração de Valores de Base de Dados no wp-config.php

```

* @since 2.6.0
*/
define('AUTH_KEY',          'II1)cW RSp}8v3# bCi>H7't *_H+2L5{'BQ+vtKg )H}UVJj.k4dLx]i@D>!!f;');
define('SECURE_AUTH_KEY',   'QGw}QAtno&^)gV5kd&$wG|?uHQ(+--TTUC4P=2GIU:6ksj]mJ5?J|{DQ=ziK~%6g<');
define('LOGGED_IN_KEY',     '2S$cE$a|8d`P[K08<CYpBa3ZcVFu#zu^LyX(x.6E$7Div|e,)7eCrmpY|Nw%T');
define('NONCE_KEY',         'pb#mdgz0buZzJ`%;g&>J-R@?_nC%x@[XJ-^~oF!0~)mF)q?{hqD__~hf|}k^=K$D');
define('AUTH_SALT',         '1@m2Pe6qnE5r{Sw7}jWM,LRVc-Hxm1|@9xssoqF$;nguwYxN]a-~+S{724je>(bu');
define('SECURE_AUTH_SALT',  '5JP759m+?y|{Cd9Ez=o/CX*(<1R+3~?Cx>!D;[N}2ot+nZwt%A0KL^N_**#~N;g?');
define('LOGGED_IN_SALT',    'RuMIUY+BaMrd9T<tZq7#k&OLPZ0~PQAVT/p_Ic|mD|ABn3LRa}fgC_0_Q?|m.rK*');
define('NONCE_SALT',       'E8_q~ZV(J+[5eN_oycc~<a.Nm[;|u;MLw3+r9J+~it-mkUD|_8Tp+eibW`sYYn#U');

/**#@-*/

```

Figura 27 - Configuração de Chaves de Segurança no wp-config.php

3.2.7 Ajuste de permissões e limpeza de ficheiros padrão

```

[ec2-user@ip-10-0-1-158 ~]$ sudo nano /usr/share/nginx/html/wp-config.php
[ec2-user@ip-10-0-1-158 ~]$ sudo nano /usr/share/nginx/html/wp-config.php
[ec2-user@ip-10-0-1-158 ~]$ sudo chown -R nginx:nginx /usr/share/nginx/html/
[ec2-user@ip-10-0-1-158 ~]$ sudo chmod -R 755 /usr/share/nginx/html/
[ec2-user@ip-10-0-1-158 ~]$ sudo rm -f /usr/share/nginx/html/index.html

```

Figura 28 - Ajuste de Permissões nginx

1) Permissões corretas no diretório do site:

- `sudo chown -R nginx:nginx /usr/share/nginx/html/`
- `sudo chmod -R 755 /usr/share/nginx/html/`

Isto garante que o utilizador nginx tenha controlo total e que o Nginx possa aceder aos ficheiros corretamente.

2) Remover a página padrão do Nginx:

- `sudo rm -f /usr/share/nginx/html/index.html`

Ao remover o index.html padrão, permitimos que o WordPress utilize o index.php como página inicial.

3.2.8 Configuração do Nginx para interpretar PHP

```

[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-1-158 ~]$ sudo nano /etc/nginx/nginx.conf

```

```
GNU nano 8.3 /etc/nginx/nginx.conf
server {
    listen      80;
    listen      [::]:80;
    server_name _;
    root        /usr/share/nginx/html;

    index index.php index.html index.htm;

    location ~ /\.php$ {
        root        /usr/share/nginx/html;
        fastcgi_pass  unix:/run/php-fpm/www.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include      fastcgi_params;
    }

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    error_page 404 /404.html;
    location = /404.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

Figura 29 - Configuração do 'location /\.php\$' no Nginx

3.2.9 Configuração do ficheiro default.conf

```
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-1-158 ~]$ sudo nano /etc/nginx/conf.d/default.conf
```

```
GNU nano 8.3 /etc/nginx/conf.d/default.conf
server {
    listen 80;
    server_name localhost;
    root /usr/share/nginx/html;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ /\.php$ {
        include fastcgi_params;
        fastcgi_pass unix:/run/php-fpm/www.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

Figura 30 - Configuração completa do servidor Nginx para WordPress

3.2.10 Teste Final e Instalação WordPress via Browser

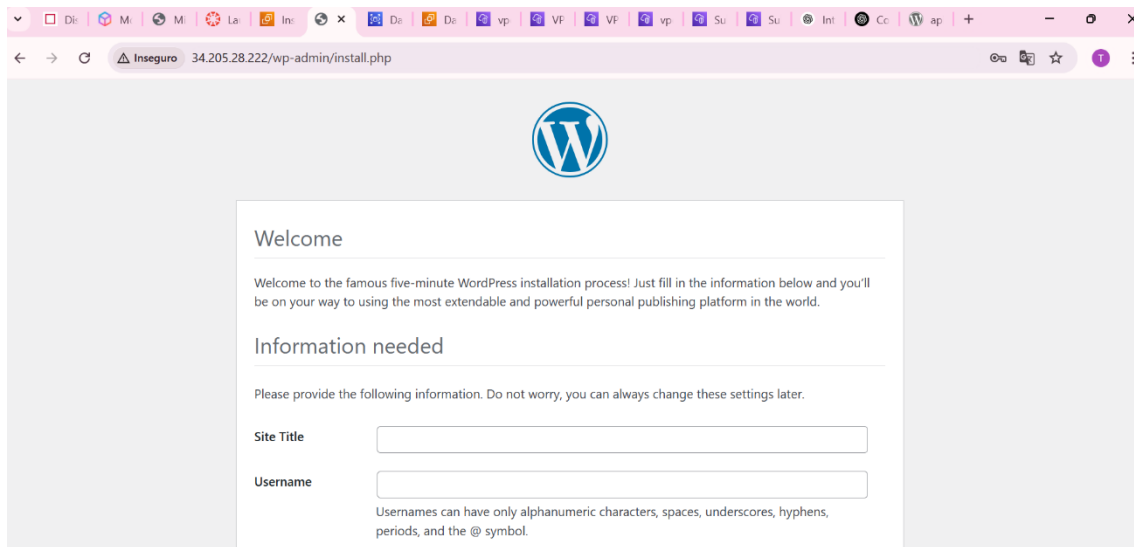
1) Verificação de Serviços

```
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl restart php-fpm
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-1-158 ~]$ sudo chown nginx:nginx /run/php-fpm/www.sock
[ec2-user@ip-10-0-1-158 ~]$ sudo chmod 660 /run/php-fpm/www.sock
[ec2-user@ip-10-0-1-158 ~]$ sudo systemctl restart php-fpm
```

2) Instalação via browser:

- Aceder a: <http://<IP-público-da-instância-WordPress>>
- Após a instalação, aceder à área de administração:

Figura 31 - Instalação do Wordpress no site



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

Username

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

Figura 32 - Log In Wordpress

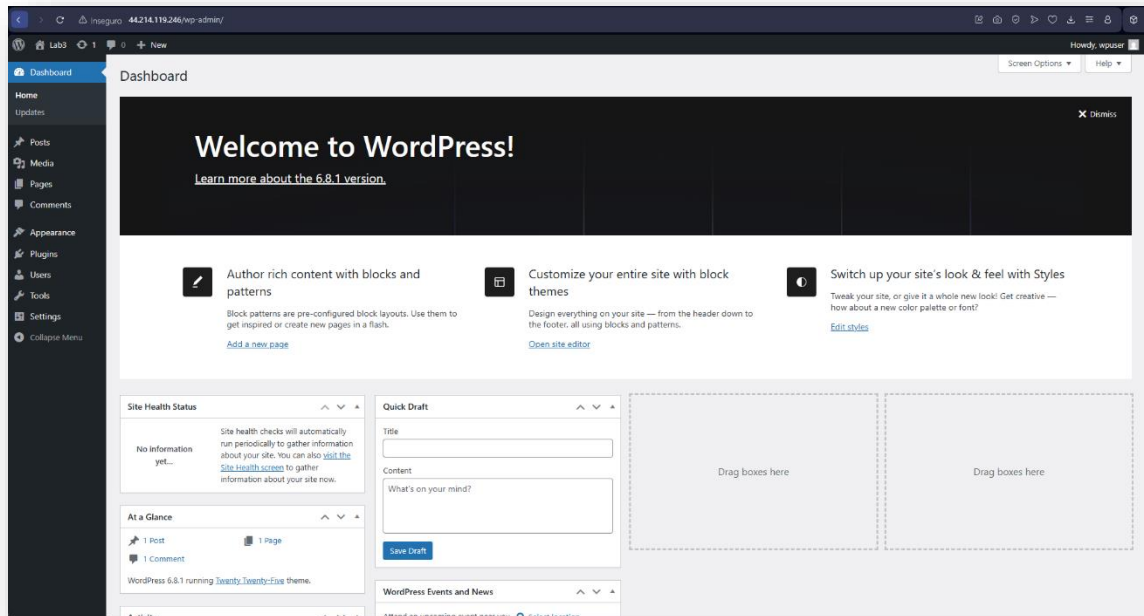


Figura 33 - Wordpress wp-admin

Conclusão

Este laboratório foi uma ótima oportunidade para aprendermos na prática como montar uma infraestrutura na AWS de forma organizada e segura. Criámos duas VPCs separadas (uma pública e outra privada) e configurámos subnets, tabelas de rotas, ligações por peering, ACLs e security groups, tudo com o objetivo de garantir uma boa separação e controlo de acessos entre os diferentes componentes.

Conseguimos lançar três instâncias EC2 com funções distintas: o Bastion Host, que funciona como ponto de entrada por SSH; o servidor WordPress, acessível pela internet; e a base de dados MySQL, colocada numa subnet privada para maior segurança. Foi interessante ver como as diferentes partes da infraestrutura se ligam entre si, por exemplo, o WordPress a comunicar com a base de dados através do IP privado, e o Bastion a permitir o acesso interno via SSH às outras instâncias.

No geral, este laboratório ajudou-nos a perceber melhor como funciona uma arquitetura típica na nuvem, com preocupações reais de segurança, acessibilidade e organização. Também ganhámos mais à-vontade com ferramentas como EC2, VPC, Route Tables, chaves SSH. Foi um desafio interessante e uma boa preparação para ambientes de trabalho mais reais.