



UNIVERSIDADE
LUSÓFONA

Sistemas de Informação na Nuvem

Relatório Laboratório 3

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	6
Route Tables Privada	7
NAT	7
Peering Connection	8
SG1 – Bastion.....	8
SG2 – WordPress.....	9
SG3 – DBInstance	10
Exercício 2 – Criar as instâncias EC2	11
Vm – Bastion	11
Vm – WordPress	12
Vm – DBInstance	13
Exercício 3 – Configuração das Instâncias	14
3.1 Configuração Instância DB:	14
3.1.1 Acesso á Instância DB a partir do Bastion.....	14
3.1.2 Instalação Maria DB	15
3.1.3 Criação da Base de Dados e Utilizador	15
3.2 Configuração Instância WordPress:	16
3.2.1 Acesso á Instância Wordpress a partir do Bastion	16
3.2.2 Instalação nginx	16
3.2.3 Instalação php	17
3.2.3 Instalação WordPress.....	18
3.2.4 Transferência dos Ficheiros do WordPress para nginx.....	18
3.2.5 Configuração do wp-config.php	19
3.2.6 Ajuste de permissões e limpeza de ficheiros padrão	20
3.2.7 Configuração do Nginx para interpretar PHP	20
3.2.8 Configuração do ficheiro default.conf	21

3.2.9	Teste Final e Instalação WordPress via Browser.....	21
	Conclusão.....	23

Introdução

O principal objetivo deste trabalho foi criar uma infraestrutura de rede segura e fiável na plataforma AWS (Amazon Web Services), composta por duas redes virtuais separadas, chamadas VPCs, que foram ligadas entre si através de uma ligação chamada *VPC Peering*. Cada uma destas redes foi organizada com duas partes: uma zona pública, onde os serviços podem ser acedidos a partir da internet, e uma zona privada, mais protegida, onde estão os serviços internos.

Para garantir a segurança, foram definidas regras rigorosas que controlam o tráfego de entrada e saída, tanto a nível das ACLs (listas de controlo de acessos) como dos *Security Groups* (grupos de segurança). Estas regras servem para permitir apenas o acesso necessário entre os diferentes componentes da rede e impedir acessos indesejados.

Foram criadas três máquinas virtuais (instâncias EC2), todas com o mesmo tipo (t2.micro) e com a mesma chave de acesso remoto (SSH), para facilitar a gestão. Cada uma destas instâncias foi colocada numa das zonas da rede. A primeira, chamada *Bastion Host*, está localizada numa subnet pública e funciona como ponto de entrada seguro: permite ligar por SSH às outras duas instâncias, que estão mais protegidas. Desta forma, nenhuma das instâncias privadas está exposta diretamente à internet, o que aumenta a segurança.

A segunda instância foi configurada com o sistema WordPress, para criar um site, e também se encontra na zona pública. Esta está acessível ao público através da internet, usando os protocolos HTTP e HTTPS. No entanto, o acesso remoto (SSH) a esta instância só é possível através da Bastion Host, garantindo maior controlo sobre quem pode aceder.

A terceira instância é um servidor de base de dados MySQL, que está na zona privada da rede. Esta não pode ser acedida diretamente da internet, estando apenas acessível a partir da instância WordPress, localizada na parte pública da outra VPC. Esta separação ajuda a proteger os dados, assegurando que só o servidor certo pode aceder à base de dados.

Foram feitos testes de ligação entre todas as máquinas e verificada a aplicação correta das regras de segurança. A infraestrutura construída cumpre os objetivos definidos e segue boas práticas para garantir a segurança e o bom funcionamento dos serviços na nuvem.

A configuração segue os princípios de uma arquitetura de rede segura e escalável. A infraestrutura pretendida está apresentada na Figura 1.

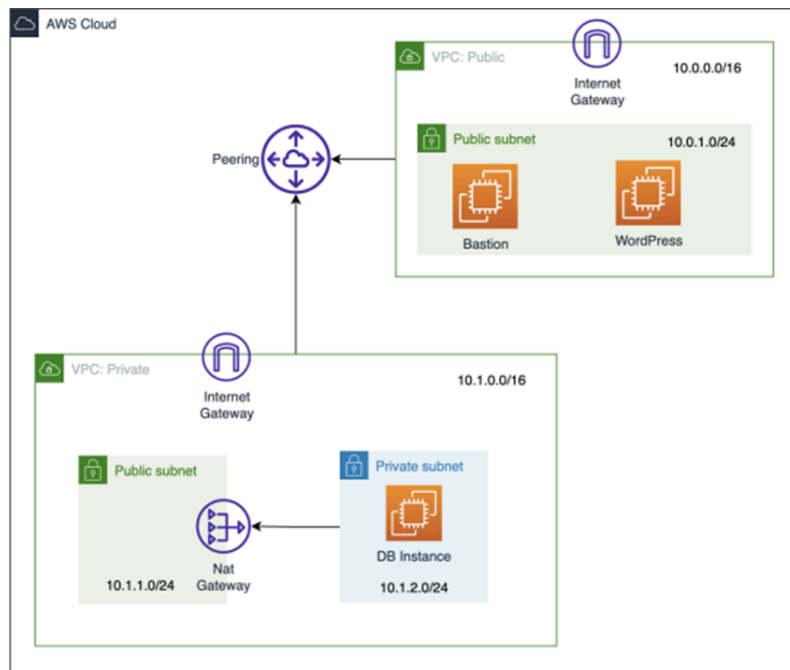


Figura 1 - Infraestrutura Objetivo

Exercício 1 – Criar infraestrutura presente na figura 1

Criação da VPC Pública e Subnets

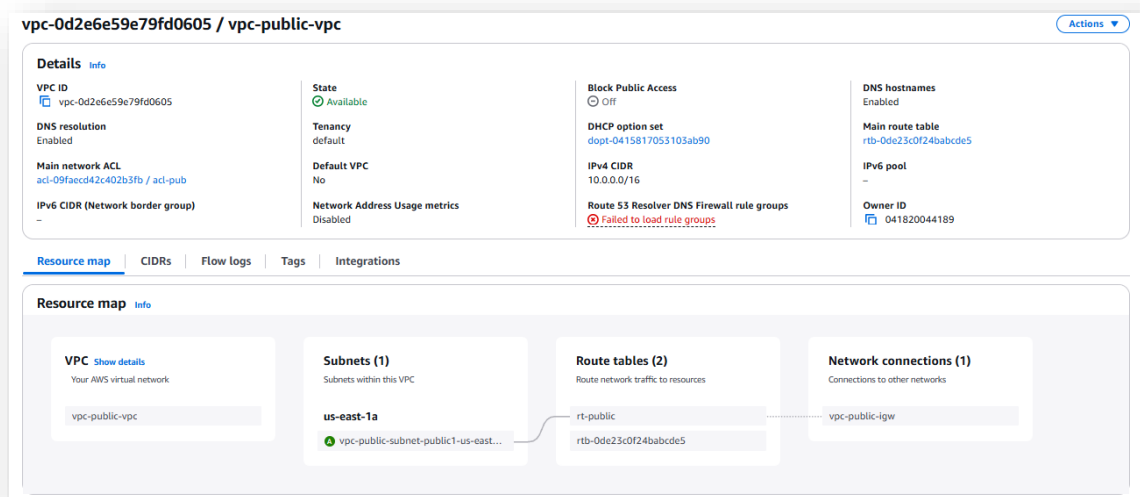


Figura 2 - VPC Publica e Subnet

Criação da VPC Privada e Subnets

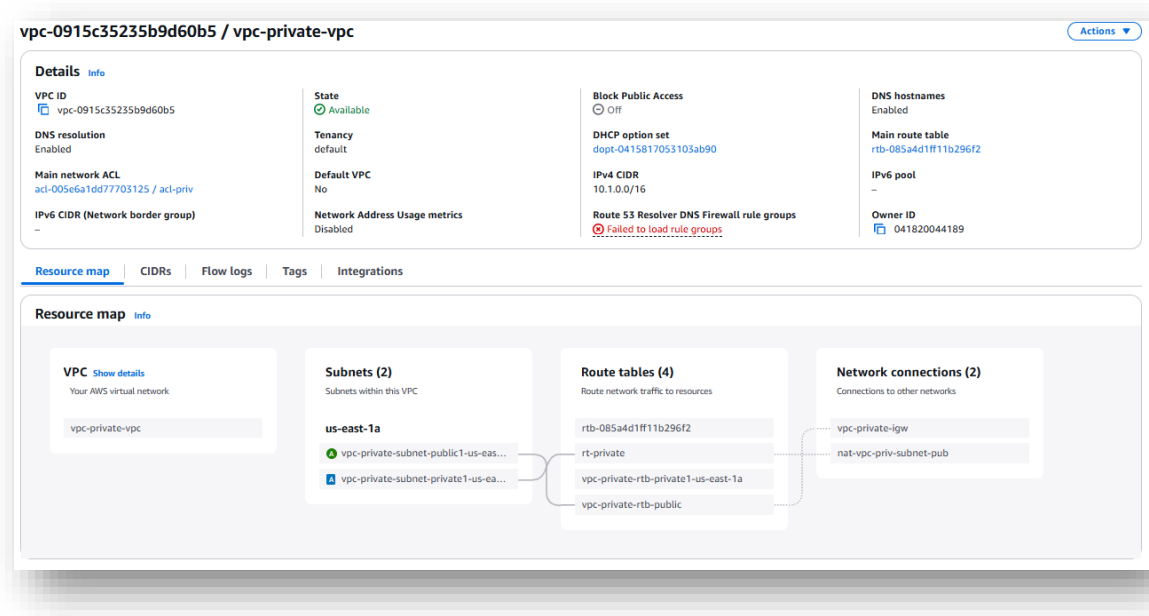


Figura 3 - VPC Privada e Subnet

Route Tables Publica

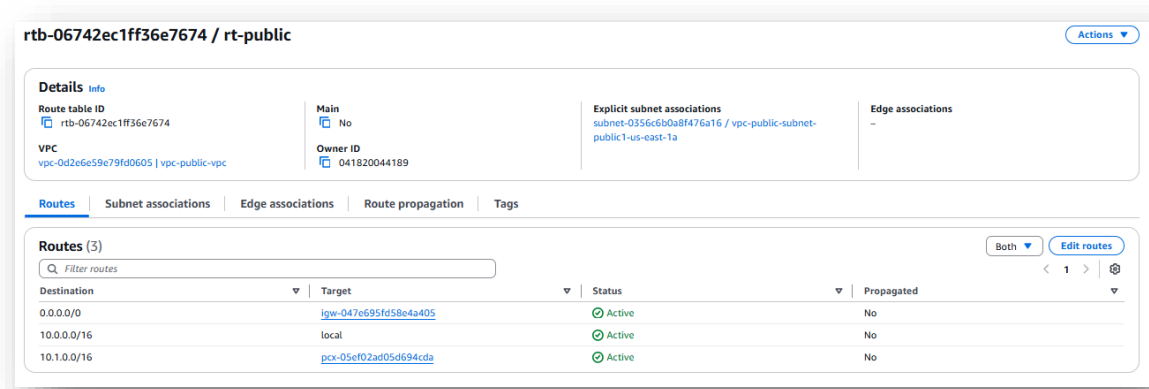


Figura 4 - Route Table Publica

Route Tables Privada

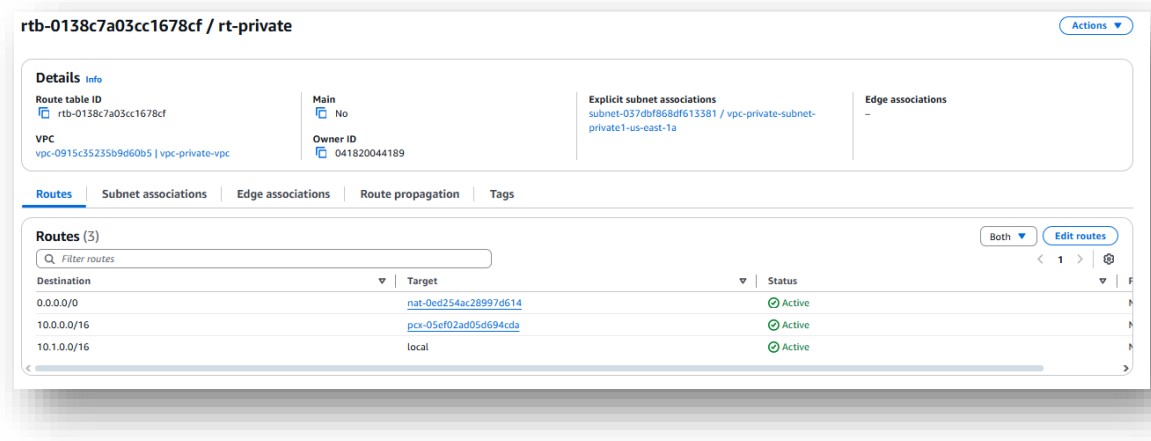


Figura 5 - Route Table Privada

NAT

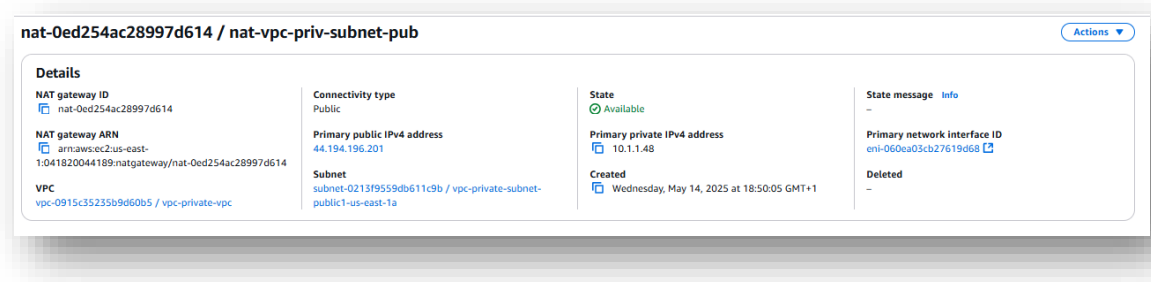


Figura 6 – NAT

Peering Connection

pcx-05ef02ad05d694cda / public-private-peering

Details

Requester owner ID

041820044189

Peering connection ID

pcx-05ef02ad05d694cda

Status

Active

Expiration time

-

Accepter owner ID

041820044189

Requester VPC

vpc-0d2e6e59e79fd0605 / vpc-public-vpc

Requester CIDRs

10.0.0.0/16

Requester Region

N. Virginia (us-east-1)

VPC Peering connection ARN

arn:aws:ec2:us-east-1:041820044189:vpc-peering-connection/pcx-05ef02ad05d694cda

Accepter VPC

vpc-0915c35235b9d60b5 / vpc-private-vpc

Accepter CIDRs

10.1.0.0/16

Accepter Region

N. Virginia (us-east-1)

DNS

Route tables

Tags

DNS settings

Requester VPC (vpc-0d2e6e59e79fd0605 / vpc-public-vpc)

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

Disabled

Accepter VPC (vpc-0915c35235b9d60b5 / vpc-private-vpc)

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

Disabled

Figura 7 - Peering Connection

SG1 – Bastion

sg-0eb96e8594ca740f9 - sg1-Bastion

Details

Security group name

sg1-Bastion

Owner

041820044189

Security group ID

sg-0eb96e8594ca740f9

Inbound rules count

2 Permission entries

Description

.

Outbound rules count

1 Permission entry

VPC ID

vpc-0d2e6e59e79fd0605

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (2)

Search

Manage tags

Edit inbound rules

	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-092b82d20174ac88f	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-0596d47bfc1b94385	IPv4	All ICMP - IPv4	ICMP	All	10.1.0.0/16	-

Figura 8 - SG1 - Bastion Inbound Rules

sg-0eb96e8594ca740f9 - sg1-Bastion Actions

Details

Security group name sg1-Bastion	Security group ID sg-0eb96e8594ca740f9	Description .	VPC ID vpc-0d2e6e59e79fd0605
Owner 041820044189	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) | [Outbound rules](#) | [Sharing - new](#) | [VPC associations - new](#) | [Tags](#)

Outbound rules (1) Manage tags Edit outbound rules

Search

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
<input type="checkbox"/>	-	sgr-06d2ded0fe45a1c54	IPv4	All traffic	All	All	0.0.0.0/0	-

Figura 9 - SG1 - Bastion Outbound Rules

SG2 – WordPress

sg-0a10b431458abe43e - sg2-WordPress Actions

Details

Security group name sg2-WordPress	Security group ID sg-0a10b431458abe43e	Description .	VPC ID vpc-0d2e6e59e79fd0605
Owner 041820044189	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) | [Outbound rules](#) | [Sharing - new](#) | [VPC associations - new](#) | [Tags](#)

Inbound rules (3) Manage tags Edit inbound rules

Search

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-02594f7484bfe2b6d	IPv4	SSH	TCP	22	10.0.1.0/24	-
<input type="checkbox"/>	-	sgr-0af20900571a71c0a	IPv4	HTTPS	TCP	443	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-08edbe6817af53a99	IPv4	HTTP	TCP	80	0.0.0.0/0	-

Figura 10 - SG2 - WordPress Inbound Rules

sg-0a10b431458abe43e - sg2-WordPress Actions

Details

Security group name sg2-WordPress	Security group ID sg-0a10b431458abe43e	Description .	VPC ID vpc-0d2e6e59e79fd0605
Owner 041820044189	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) | [Outbound rules](#) | [Sharing - new](#) | [VPC associations - new](#) | [Tags](#)

Outbound rules (1) Manage tags Edit outbound rules

Search

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
<input type="checkbox"/>	-	sgr-00c713e827a65135c	IPv4	All traffic	All	All	0.0.0.0/0	-

Figura 11 - SG2 - WordPress Outbound Rules

SG3 – DBInstance

sg-0da99090992999a6b - sg3-DBInstance

Details

Security group name
sg3-DBInstance

Owner
041820044189

Security group ID
sg-0da99090992999a6b

Inbound rules count
3 Permission entries

Description
-

Outbound rules count
1 Permission entry

VPC ID
vpc-0915c35235b9d60b5

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Inbound rules (3)

Manage tags

Edit inbound rules

Search

	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-088e3253ca57ba5d7	IPv4	MySQL/Aurora	TCP	3306	10.0.0.0/16	-
<input type="checkbox"/>	-	sgr-066de98baabf90afd	IPv4	All ICMP - IPv4	ICMP	All	10.0.0.0/16	-
<input type="checkbox"/>	-	sgr-0592da089a1d3ec8	IPv4	SSH	TCP	22	10.0.0.0/16	-

Figura 12 - SG3 - DBInstance Inbound Rules

sg-0da99090992999a6b - sg3-DBInstance

Details

Security group name
sg3-DBInstance

Owner
041820044189

Security group ID
sg-0da99090992999a6b

Inbound rules count
3 Permission entries

Description
-

Outbound rules count
1 Permission entry

VPC ID
vpc-0915c35235b9d60b5

Inbound rules

Outbound rules

Sharing - new

VPC associations - new

Tags

Outbound rules (1)

Manage tags

Edit outbound rules

Search

	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
<input type="checkbox"/>	-	sgr-0f527c6b763b6c134	IPv4	All traffic	All	All	0.0.0.0/0	-

Figura 13 - SG3 - DBInstance Outbound Rules

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

[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

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**

lab3KeyPair

Create new key pair

▼ Network settings Info

VPC - **required** Info

vpc-0d2e6e59e79fd0605 (vpc-public-vpc)
10.0.0.0/16

Subnet Info

subnet-0356c6b0a8f476a16 vpc-public-subnet-public1-us-east-1a
VPC: vpc-0d2e6e59e79fd0605 Owner: 041820044189 Availability Zone: us-east-1a
Zone type: Availability Zone IP addresses available: 249 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 sg-0eb968594ca740f9 X
VPC: vpc-0d2e6e59e79fd0605

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

▼ Summary

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...[read more](#)
ami-0953476d60561c955

Virtual server type (instance type)

t2.micro

Firewall (security group)

sg1-Bastion

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 GB of bandwidth to the internet.

Cancel

Launch instance

Preview code

▼ Summary

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...[read more](#)
ami-0953476d60561c955

Virtual server type (instance type)

t2.micro

Firewall (security group)

sg1-Bastion

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 GB of bandwidth to the internet.

Cancel

Launch instance

Preview code

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

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

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

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

lab3KeyPair

Create new key pair

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0d2e6e59e79fd0605 (vpc-public-vpc)

10.0.0.0/16

Create new VPC

Subnet [Info](#)

subnet-0356c6b0a8f476a16

vpc-public-subnet-public1-us-east-1a

VPC: vpc-0d2e6e59e79fd0605 Owner: 041820044189 Availability Zone: us-east-1a

Zone type: Availability Zone IP addresses available: 248 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 sg-0a10b431458abe43e X

VPC: vpc-0d2e6e59e79fd0605

Compare security group rules

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

► Advanced network configuration

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...read more

ami-0953476d60561c955

Virtual server type (instance type)

t2.micro

Firewall (security group)

sg2-WordPress

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.

Cancel

Launch instance

Preview code

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...read more

ami-0953476d60561c955

Virtual server type (instance type)

t2.micro

Firewall (security group)

sg2-WordPress

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.

Cancel

Launch instance

Preview code

Figura 15 - Vm - WordPress

Vm – DBInstance

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-DBInstance

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)

Free tier eligible

Virtualization: hvm

ENA enabled: true

Root device type: ebs

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

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

lab3KeyPair

Create new key pair

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0915c35235b9d60b5 (vpc-private-vpc)

10.1.0.0/16

Subnet [Info](#)

subnet-0213f9559db611c9b

VPC: vpc-0915c35235b9d60b5 Owner: 041820044189 Availability Zone: us-east-1a

Zone type: Availability Zone IP addresses available: 250 CIDR: 10.1.1.0/24

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

sg3-DBInstance sg-0da9909092999a6b

VPC: vpc-0915c35235b9d60b5

Compare security group rules

► Advanced network configuration

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.7.2...read more

ami-0953476d60561c955

Virtual server type (instance type)

t2.micro

Firewall (security group)

sg3-DBInstance

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

Figura 16 - Vm - DBInstance

Exercício 3 – Configuração das Instâncias

3.1 Configuração Instância DB:

3.1.1 Acesso á Instância DB a partir do Bastion

```
C:\Users\André Jesus>cd Downloads

C:\Users\André Jesus\Downloads>scp -i "lab3KeyPair.pem" lab3KeyPair.pem ec2-user@34.205.85.182:.\ssh
The authenticity of host '34.205.85.182 (34.205.85.182)' can't be established.
ED25519 key fingerprint is SHA256:CnokyVx1lYqrIdkoMKUhbSa6GjVhIbBt1JgEdkg7z6E.
This host key is known by the following other names/addresses:
C:\Users\André Jesus/.ssh/known_hosts:8: 3.227.232.16
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '34.205.85.182' (ED25519) to the list of known hosts.
lab3KeyPair.pem                                     100% 1674   13.5KB/s   00:00

C:\Users\André Jesus\Downloads>ssh -i "lab3KeyPair.pem" ec2-user@ec2-34-205-85-182.compute-1.amazonaws.com
The authenticity of host 'ec2-34-205-85-182.compute-1.amazonaws.com (34.205.85.182)' can't be established.
ED25519 key fingerprint is SHA256:CnokyVx1lYqrIdkoMKUhbSa6GjVhIbBt1JgEdkg7z6E.
This host key is known by the following other names/addresses:
C:\Users\André Jesus/.ssh/known_hosts:8: 3.227.232.16
C:\Users\André Jesus/.ssh/known_hosts:10: 34.205.85.182
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-205-85-182.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\  #####
~~ \#####\
~~  \###|
~~   \#/  ---
~~      V~'  '---> https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-1-162 ~]$ ls -l
total 8
-rw-rw-r--. 1 ec2-user ec2-user 1674 May 14 21:27 lab3KeyPair.pem
-rw-rw-r--. 1 ec2-user ec2-user 1674 May 16 09:32 ssh
```

Figura 17 - Acesso ao Bastion

```
[ec2-user@ip-10-0-1-162 ~]$ ssh -i "lab3KeyPair.pem" ec2-user@10.1.1.149
The authenticity of host '10.1.1.149 (10.1.1.149)' can't be established.
ED25519 key fingerprint is SHA256:Eh6FPK6IvmPeVnWvAtIOZV00wgaSJA3HDA02AwGt/NE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.1.1.149' (ED25519) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!                @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'lab3KeyPair.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "lab3KeyPair.pem": bad permissions
ec2-user@10.1.1.149: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-1-162 ~]$ sudo ssh -i "lab3KeyPair.pem" ec2-user@10.1.1.149
The authenticity of host '10.1.1.149 (10.1.1.149)' can't be established.
ED25519 key fingerprint is SHA256:Eh6FPK6IvmPeVnWvAtIOZV00wgaSJA3HDA02AwGt/NE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.1.1.149' (ED25519) to the list of known hosts.

#_
~\  #####
~~ \#####\
~~  \###|
~~   \#/  ---
~~      V~'  '---> https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-1-1-149 ~]$
```

Figura 18 - Acesso á instância DB a partir do Bastion

3.1.2 Instalação Maria DB

```
[ec2-user@ip-10-1-162 ~]$ sudo ssh -i "lab3KeyPair.pem" ec2-user@10.1.1.149  
#_  
#####  
Amazon Linux 2023  
#####  
###|  
#/ --  
V~' i->  
  
Last login: Fri May 16 09:52:35 2025 from 10.0.1.162  
[ec2-user@ip-10-1-149 ~]$ sudo yum update -y  
Amazon Linux 2023 repository                                61 MB/s | 38 MB      00:00  
  
Amazon Linux 2023 Kernel Livepatch repository              112 kB/s | 16 kB     00:00  
Dependencies resolved.  
Nothing to do.  
Complete!  
[ec2-user@ip-10-1-149 ~]$ sudo dnf update  
Last metadata expiration check: 0:05:00 ago on Fri May 16 11:04:05 2025.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[ec2-user@ip-10-1-149 ~]$ sudo dnf install mariadb105-server  
Last metadata expiration check: 0:05:16 ago on Fri May 16 11:04:05 2025.  
Dependencies resolved.
```

Figura 19 - Instalação MariaDB

3.1.3 Criação da Base de Dados e Utilizador

```
[ec2-user@ip-10-1-1-149 ~]$ mysql -u root -p
Enter password:
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
[ec2-user@ip-10-1-1-149 ~]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.5.25-MariaDB MariaDB Server

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)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'wpuser'@'10.0.0.%' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'10.0.0.%';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT;
Bye
```

Figura 20 - Criação BD Wordpress e wpuser

3.2 Configuração Instância WordPress:

3.2.1 Acesso á Instância Wordpress a partir do Bastion

```
C:\Users\André Jesus>cd Downloads
C:\Users\André Jesus\Downloads>ssh -i "lab3KeyPair.pem" ec2-user@ec2-34-201-48-100.compute-1.amazonaws.com

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sat May 17 14:52:10 2025 from 81.84.119.19
[ec2-user@ip-10-0-1-162 ~]$ sudo ssh -i "lab3KeyPair.pem" ec2-user@10.0.1.158
The authenticity of host '10.0.1.158 (10.0.1.158)' can't be established.
ED25519 key fingerprint is SHA256:5HoDXqecPlmFpQ0LSxhE9yDnPJauh3i6HIJ4UvaSw2A.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.158' (ED25519) to the list of known hosts.

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-1-158 ~]$ ls
```

Figura 21 - Acesso á Instacia Wordpress a partir do Bastion

3.2.2 Instalação nginx

```
[ec2-user@ip-10-0-1-158 ~]$ sudo yum install -y nginx
Amazon Linux 2023 Kernel Livepatch repository                               164 kB/s | 16 kB    00:00
Dependencies resolved.
=====
Package                                Architecture      Version                                Repository      Size
=====
Installing:
nginx                                   x86_64            1:1.26.3-1.amzn2023.0.1               amazonlinux     33 k
Installing dependencies:
generic-logos-httpd                   noarch            18.0.0-12.amzn2023.0.3                amazonlinux     19 k
gperftools-libs                       x86_64            2.9.1-1.amzn2023.0.3                  amazonlinux    308 k
libunwind                             x86_64            1.4.0-5.amzn2023.0.2                  amazonlinux     66 k
nginx-core                             x86_64            1:1.26.3-1.amzn2023.0.1               amazonlinux    670 k
nginx-filesystem                      noarch            1:1.26.3-1.amzn2023.0.1               amazonlinux     9,6 k
nginx-mimetypes                       noarch            2.1.49-3.amzn2023.0.3                 amazonlinux     21 k
Transaction Summary
=====
Install 7 Packages
```

Figura 22 - 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 23 - Inicialização nginx

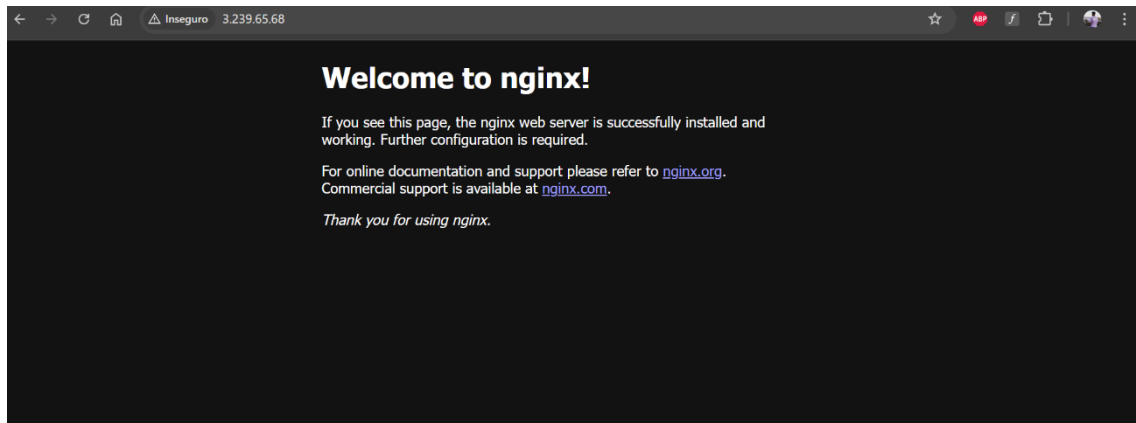


Figura 24 - Acessar a página principal local do servidor para verificar a instalação do nginx

3.2.3 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 25 - Configuração do Nginx para WordPress

3.2.5 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', 'wpuser' );  
  
/** Database password */  
define( 'DB_PASSWORD', '1234' );  
  
/** Database hostname */  
define( 'DB_HOST', '10.1.1.149' );
```

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$7Div|e,)7eCrmpY|NeW%T');  
define('NONCE_KEY', 'pb#mdgz0buZzJ`%;g&>J-R@?_nC%x@[XJ-^~oF!O~)mF)q?{hqD__~hf|)k^=K$D');  
define('AUTH_SALT', '1@m2Pe6qnE5r{Sw7)jWM,LRVc-Hxm1|@9xssqF$;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&0LPZ0~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

3.2.6 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.

3.2.7 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

3.2.8 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

3.2.9 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:

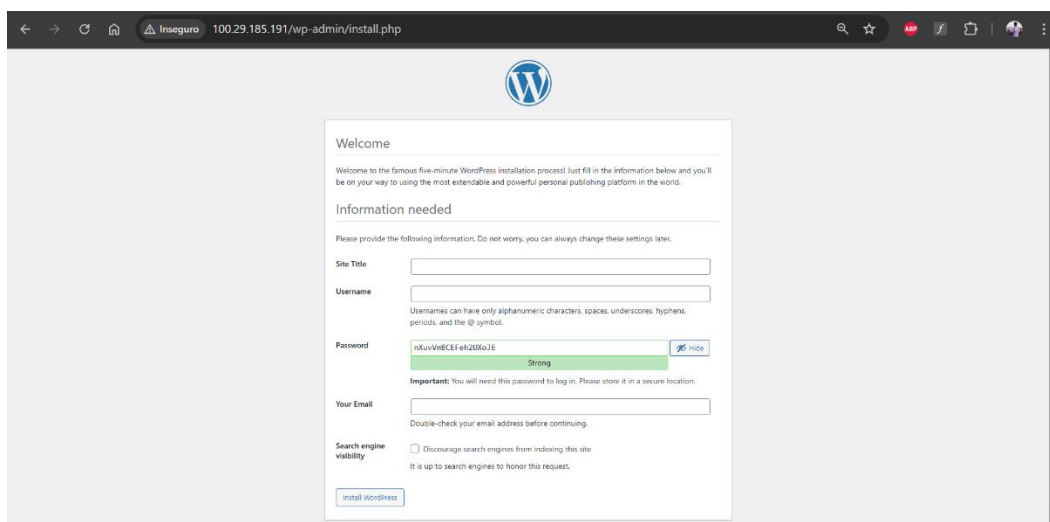


Figura 31 - Instalação do Wordpress no site



Figura 32 - Log In Wordpress

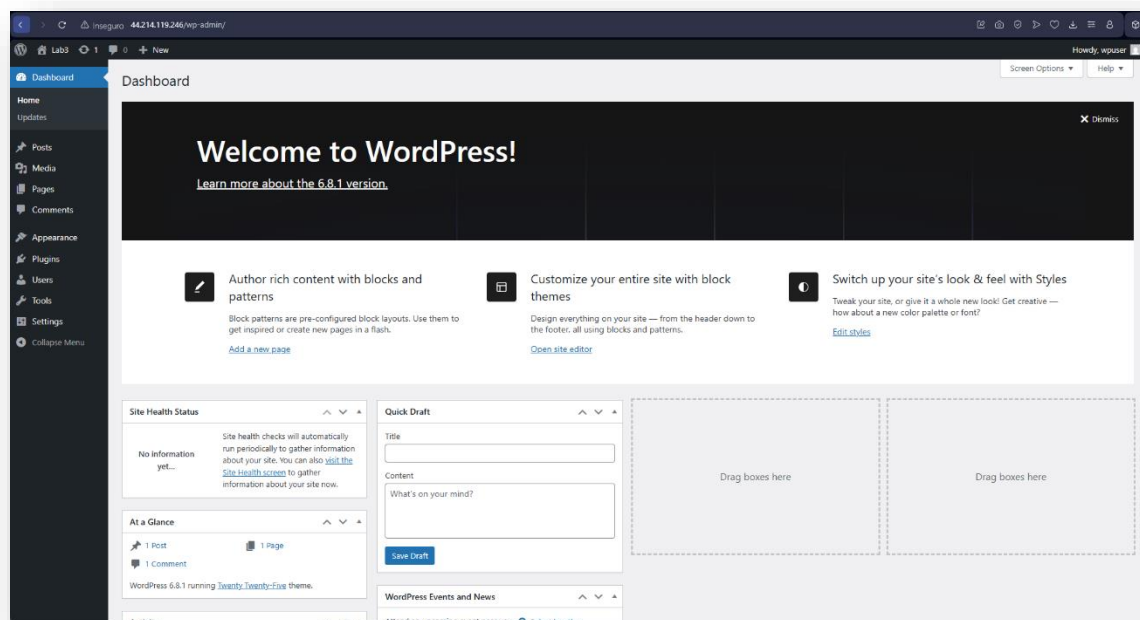


Figura 33 - Wordpress wp-admin

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.