UNIVERSIDADE
LUSÓFONA

Sistemas de Informação na Nuvem

Relatório Laboratório 2

Tomás Nave, a22208623

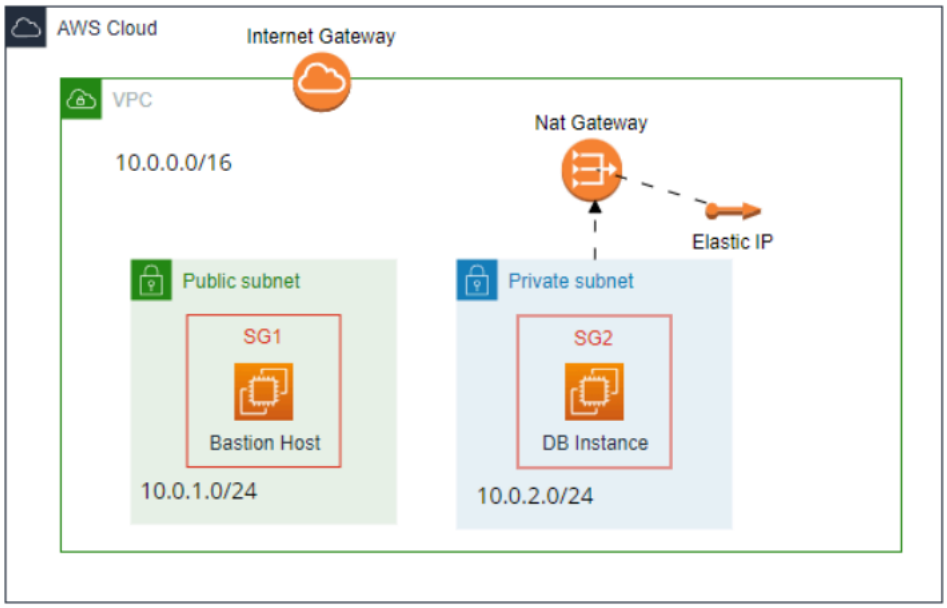
André Jesus, a22207061

Índice

Índice 2
Introdução 3
Exercício 1 – Criar infraestrutura presente na figura 1 3
1.1 Criação da VPC e Subnets 3
1.2 Criação de NAT Gateway associada à private subnet 4
1.3 Definição de ACLs 5
Exercício 2 – Criar as instâncias EC2 7
2.1 Criação de instância Bastion Host 7
2.2 Criação de instância DB instance 8
Exercício 3 – Criação de Security Groups 10
3.1 Criação de Security Group relativo ao Bastion Host (SG1) 10
3.2 Criação de Security Group relativo DB instance (SG2) 11
Exercício 4 – Testes – Validação da Infraestrutura 12
Conclusão 15

Introdução

Este relatório documenta os passos realizados no Laboratório 2, cujo objetivo foi criar uma infraestrutura segura na AWS utilizando VPC, subnets públicas e privadas, regras de acesso, instâncias EC2, e uma configuração de Bastion Host para acesso controlado. A configuração segue os princípios de uma arquitetura de rede segura e escalável. A infraestrutura pretendida está apresentada na Figura 1.

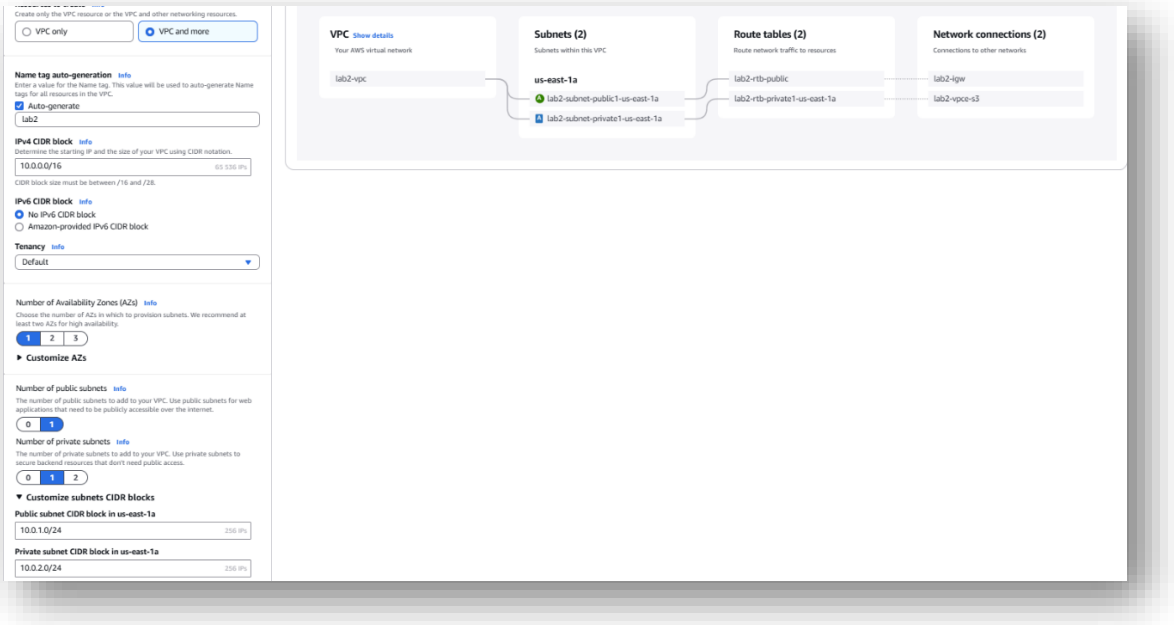


Exercício 1 – Criar infraestrutura presente na figura 1

1.1 Criação da VPC e Subnets

Passo 1: Criação VPC

No primeiro passo, foi criada uma Virtual Private Cloud (VPC) com o nome "lab2-vpc", utilizando o IPv4 10.0.0.0/16. A VPC foi configurada para permitir a criação de subnets públicas e privadas.



Passo 2: Criação das Subnets

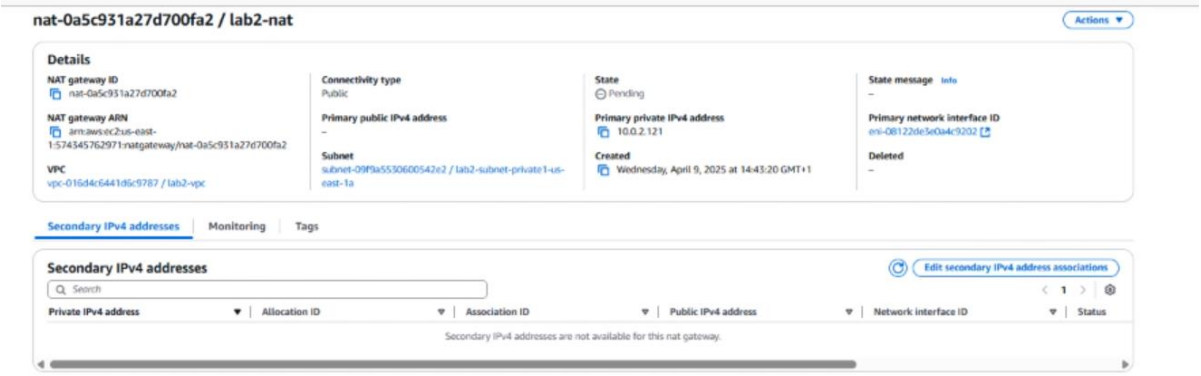
Após a criação da VPC, foi realizado o processo de configuração de duas subnets:

- A **public subnet** com o bloco CIDR 10.0.1.0/24 foi destinada a hospedar a instância Bastion Host, permitindo o acesso à internet através de um IP público.
- A **private subnet** com o bloco CIDR 10.0.2.0/24 foi configurada para a instância DB, de modo que não tenha acesso direto à internet, assegurando maior segurança para a base de dados.

1.2 Criação de NAT Gateway associada à private subnet

Passo 1: Criação NAT Gateway

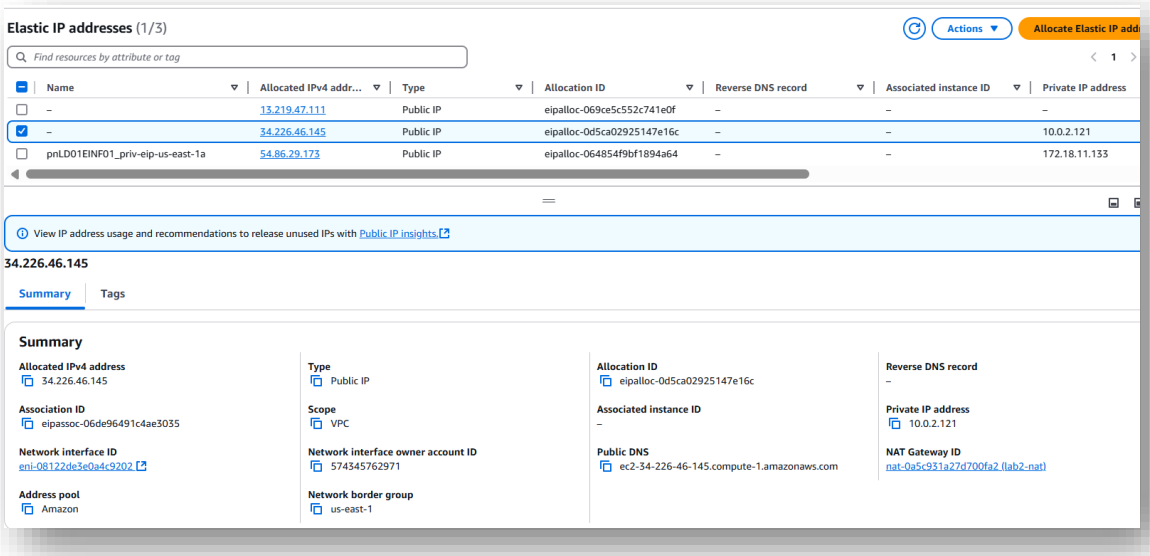
Neste passo, foi criado um NAT Gateway, que foi colocado na public subnet, mas foi associado à private subnet para possibilitar que a instância DB (na private subnet) tenha acesso à internet.



Passo 2: Elastic IP

O NAT Gateway foi associado a um **Elastic IP**, que fornece um IP estático acessível pela internet.

O NAT Gateway permite que a instância na private subnet faça requisições externas (como atualizações de pacotes ou instalação de softwares), sem precisar expor a instância diretamente à internet.



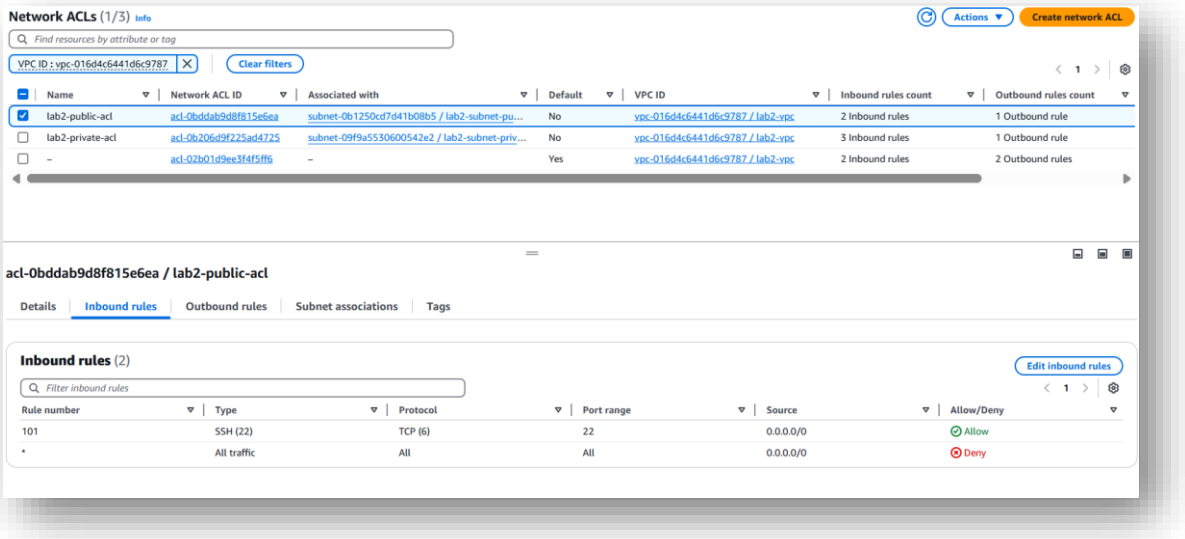
1.3 Definição de ACLs

Para garantir a segurança da infraestrutura, foram configuradas **Access Control Lists (ACLs)** para controlar o tráfego nas subnets pública e privada. Estas acls foram configuradas de forma a estarem coerentes com os Security Groups criados posteriormente.

ACL da Public Subnet

Esta NACL destina-se à subnet onde se encontra o Bastion Host, permitindo acesso direto a partir da internet:

- **Inbound Rules:**
    - Permitir **SSH (porta 22)** a partir de **qualquer origem (0.0.0.0/0)**.
- **Outbound Rules:**
    - Permitir **todo o tráfego de saída (0.0.0.0/0)**, garantindo conectividade externa.

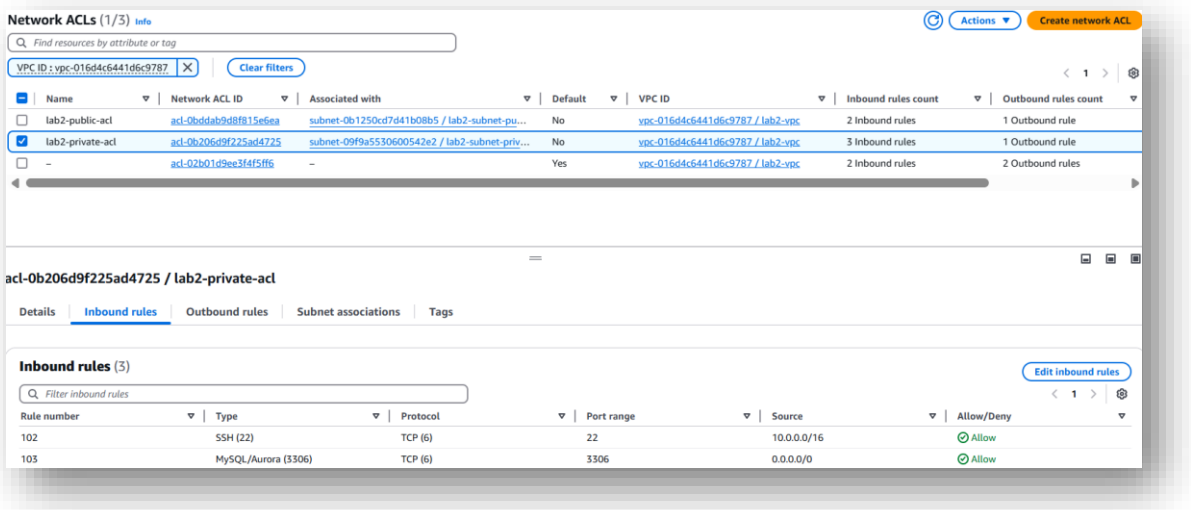


ACL da Private Subnet

A NACL da subnet privada foi configurada para permitir apenas comunicações essenciais com o exterior e com a subnet pública:

- **Inbound Rules:**
    - Permitir **SSH (porta 22)** a partir de toda a **VPC (10.0.0.0/16)**, possibilitando o acesso interno entre instâncias.
    - Permitir **MySQL/Aurora (porta 3306)** a partir de **qualquer origem (0.0.0.0/0)**.
      Nota: Apesar de esta regra permitir tráfego de qualquer origem, a instância de base de dados está protegida pois está alojada numa **subnet privada**, sem acesso direto a partir da internet. Esta parte está também melhor explicada na secção **Exercício 3 – Criação de Security Groups**

- **Outbound Rules:**
    - Permitir **todo o tráfego de saída (0.0.0.0/0)**, necessário para que a instância da base de dados possa aceder à internet através do NAT Gateway (por exemplo, para instalar pacotes).



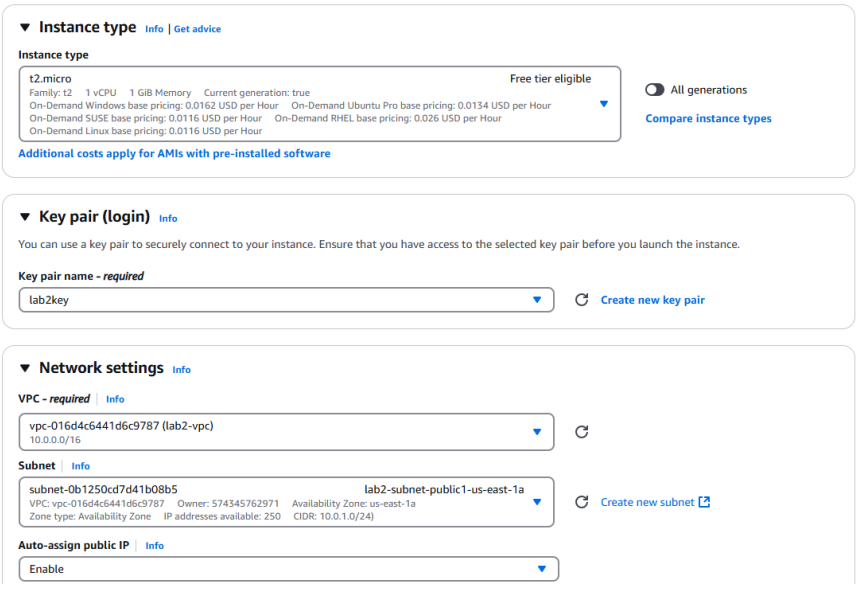
Exercício 2 – Criar as instâncias EC2

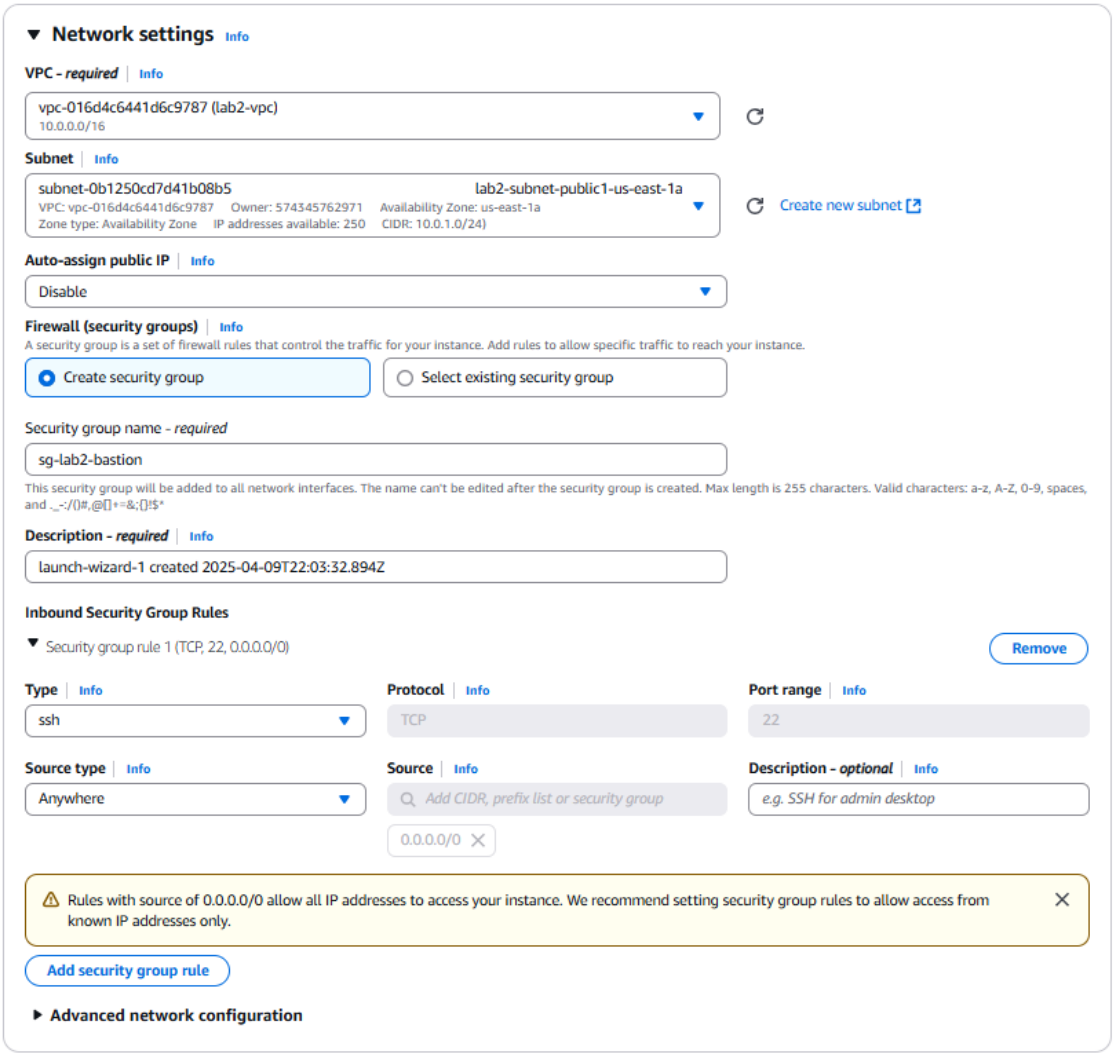
2.1 Criação de instância Bastion Host

Foi criada uma instância **t2.micro** na **public subnet**, devidamente associada à **VPC** configurada no exercício anterior.

As configurações principais incluem:

- Utilização de um **par de chaves (key pair)** para acesso via **SSH**.
- Associação ao **Security Group SG1**, configurado para permitir acesso externo apenas na **porta 22 (SSH)**.
- Seleção da **public subnet (10.0.1.0/24)** como zona de implementação.
- Atribuição automática de **IP público**, permitindo comunicação com o exterior.

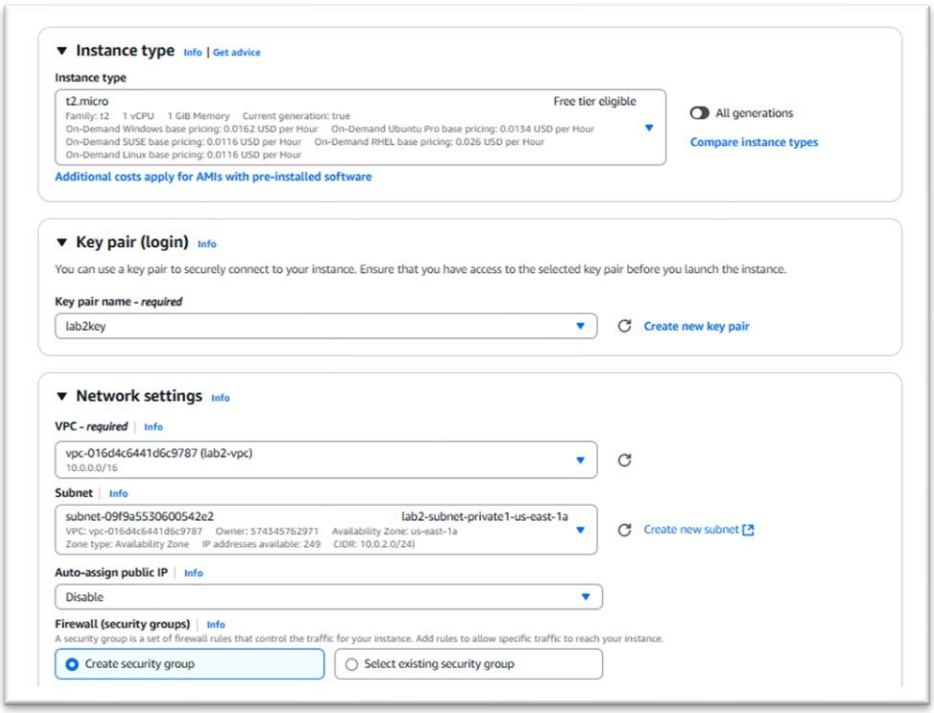


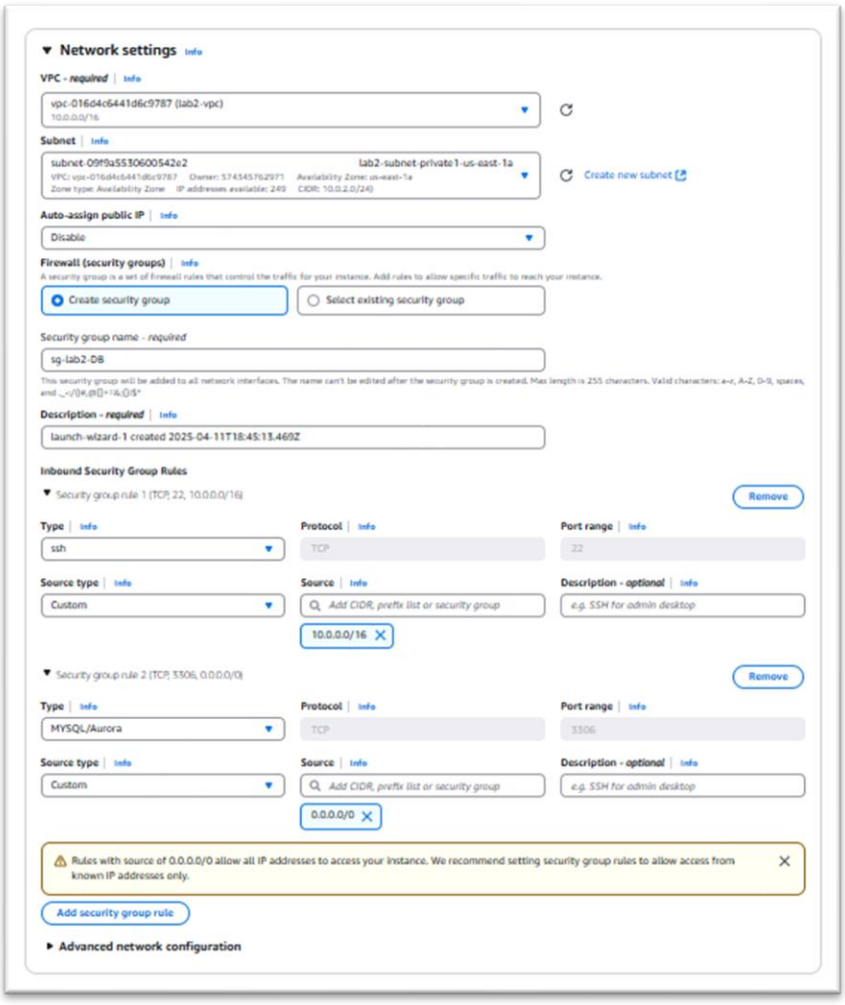


2.2 Criação de instância DB instance

Foi criada uma instância **t2.micro** na **private subnet**, também associada à mesma **VPC**. Esta instância representa a base de dados e foi configurada com os seguintes parâmetros:

- Utilização do **mesmo par de chaves** usado na instância Bastion Host, permitindo acesso indireto via SSH.
- Associação ao **Security Group SG2**, o qual permite apenas:
    - Conexões na **porta 3306 (MySQL)** provenientes do grupo SG1 (ou seja, da instância Bastion Host).
- Acesso à internet é garantido **exclusivamente via NAT Gateway**.
- A instância foi criada na **private subnet (10.0.2.0/24)**, sem IP público atribuído.


Instance type Info | Get advice
Instance type
t2.micro
Free tier eligible
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
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
lab2key
Create new key pair
Network settings Info
VPC - required | Info
vpc-016d4c6441d6c9787 (lab2-vpc)
10.0.0.0/16
Subnet | Info
subnet-09f9a5530600542e2
lab2-subnet-private1-us-east-1a
VPC: vpc-016d4c6441d6c9787 Owner: 574345762971 Availability Zone: us-east-1a
Zone type: Availability Zone IP addresses available: 249 CIDR: 10.0.2.0/24)
Create new subnet
Auto-assign public IP | Info
Disable
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

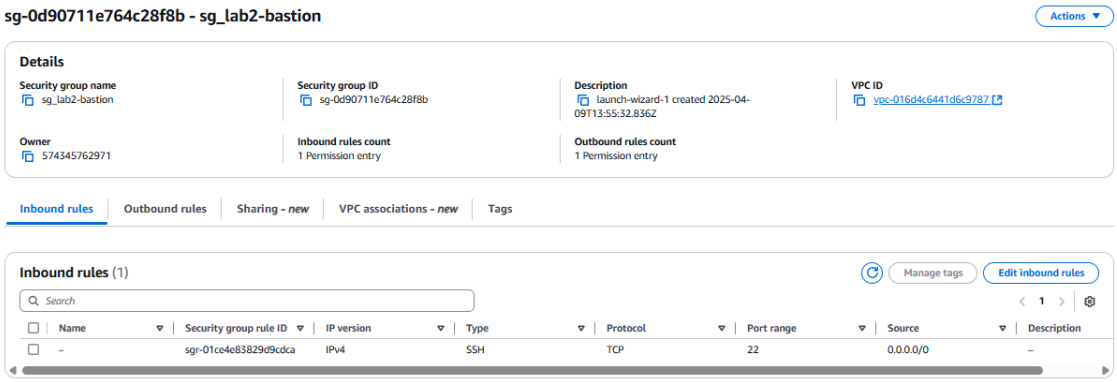

Network settings Info
VPC - required | Info
vpc-016d4c6441d6c9787 (lab2-vpc)
10.0.0.0/16
Subnet | Info
subnet-09f9a5530600542e2
lab2-subnet-private1-us-east-1a
Create new subnet
Auto-assign public IP | Info
Disable
Firewall (security groups) | Info
A security group is a set of firewall rules that control 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
sg-lab2-DB
Description - required | Info
launch-wizard-1 created 2025-04-11T18:45:13.469Z
Inbound Security Group Rules
Security group rule 1 (TCP, 22, 10.0.0.0/16)
Remove
Type | Info
ssh
Protocol | Info
TCP
Port range | Info
22
Source type | Info
Custom
Source | Info
Add CIDR, prefix list or security group
Description - optional | Info
e.g. SSH for admin desktop
10.0.0.0/16
Security group rule 2 (TCP, 3306, 0.0.0.0/0)
Remove
Type | Info
MYSQL/Aurora
Protocol | Info
TCP
Port range | Info
3306
Source type | Info
Custom
Source | Info
Add CIDR, prefix list or security group
Description - optional | Info
e.g. SSH for admin desktop
0.0.0.0/0
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
Advanced network configuration

Exercício 3 – Criação de Security Groups

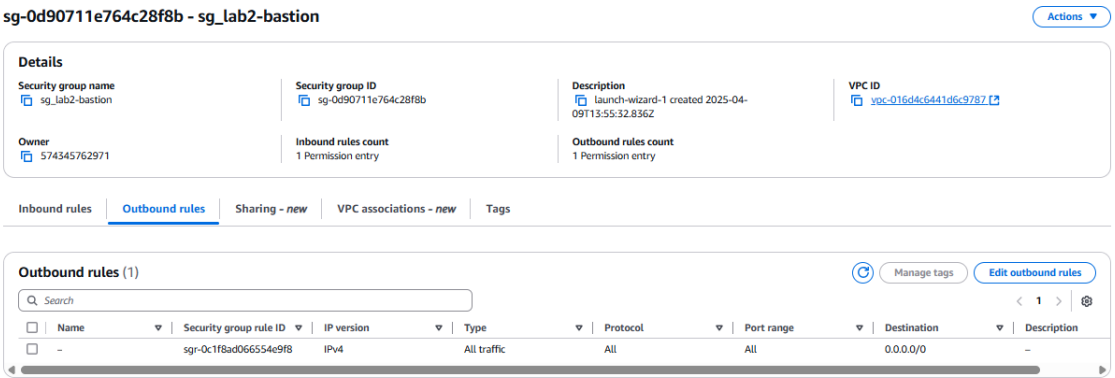
3.1 Criação de Security Group relativo ao Bastion Host (SG1)

O Security Group SG1 foi criado com o objetivo de garantir acesso seguro à instância Bastion Host, que serve como ponto de entrada para a VPC. As configurações foram as seguintes:

- **Inbound rules**:
  - Porta **22 (SSH)**: aberta apenas para o endereço IP público do utilizador (ou gama restrita de IPs), de forma a limitar o acesso externo e mitigar riscos de ataque por força bruta.



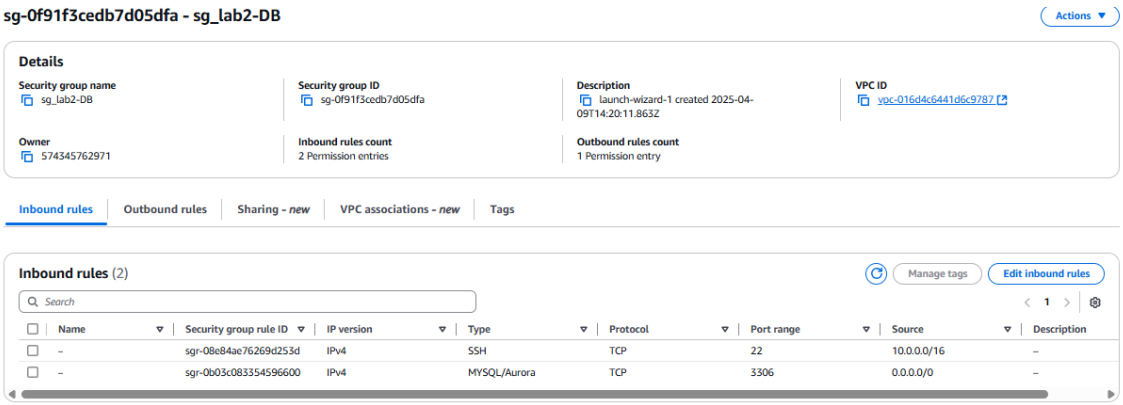
- **Outbound rules**:
  - Permissão para **todo o tráfego de saída** (0.0.0.0/0), permitindo que a instância Bastion Host possa estabelecer comunicações com o exterior, por exemplo, para atualizações de sistema ou instalação de pacotes via repositórios online.



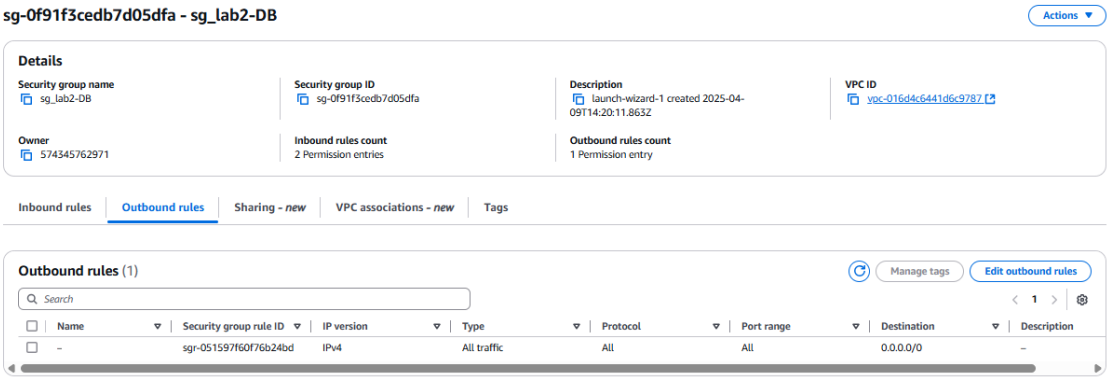
3.2 Criação de Security Group relativo DB instance (SG2)

O Security Group SG2 foi configurado para proteger a instância de base de dados localizada na subnet privada. Esta instância não deve estar acessível a partir da internet uma vez que não possui IP público e encontra-se isolada numa subnet privada com ligação de saída apenas via NAT Gateway. Posto isto o seu acesso é exclusivamente permitido a partir da VPC, mais concretamente, da instância Bastion Host.

- **Inbound rules**:
  Porta **3306 (MySQL/Aurora)**: aceita tráfego proveniente de qualquer origem (**0.0.0.0/0**).
  Nota: Apesar de a origem ser ampla, a instância está protegida por estar numa subnet privada, sem possibilidade de acesso externo direto. Esta configuração permite acesso interno desde que o tráfego consiga atingir a subnet privada — o que, neste caso, só é possível a partir da VPC.
  Esta abordagem garante flexibilidade de acesso interno (por exemplo, para ferramentas de monitorização ou outras instâncias na VPC), mantendo a segurança a nível da arquitetura de rede.



- **Outbound rules**:
  - Permissão para **todo o tráfego de saída** (0.0.0.0/0), o que é necessário para que a instância possa, por exemplo, descarregar atualizações ou comunicar com serviços externos através da NAT Gateway.



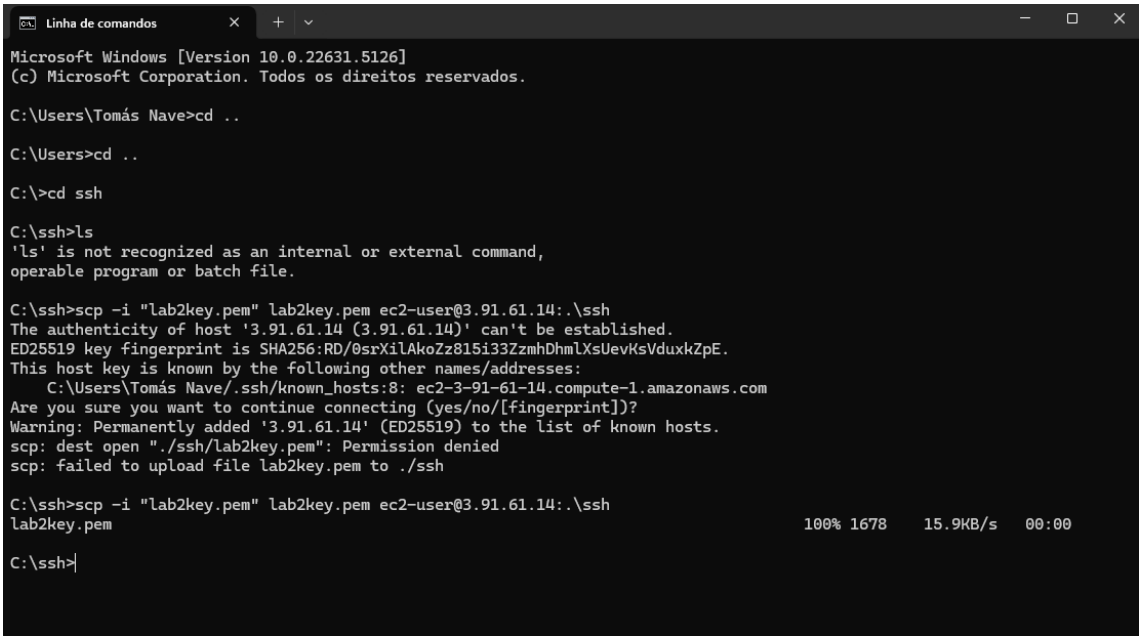
Esta configuração garante o isolamento da instância de base de dados do tráfego externo, mantendo-a acessível apenas através do Bastion Host, conforme as boas práticas de segurança em ambientes cloud.

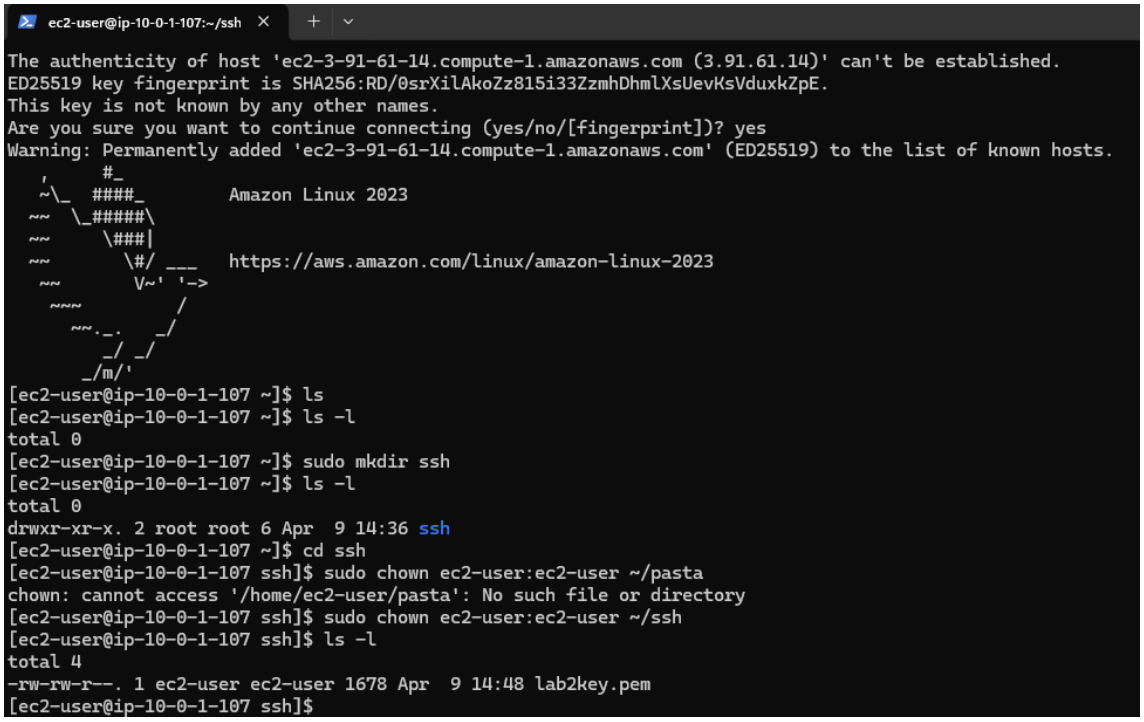
Exercício 4 – Testes – Validação da Infraestrutura

Após a criação de toda a infraestrutura, foram realizados testes práticos nas instâncias EC2 com o objetivo de validar a conectividade e garantir que a arquitetura está funcional e segura.

Acesso ao Bastion Host via SSH

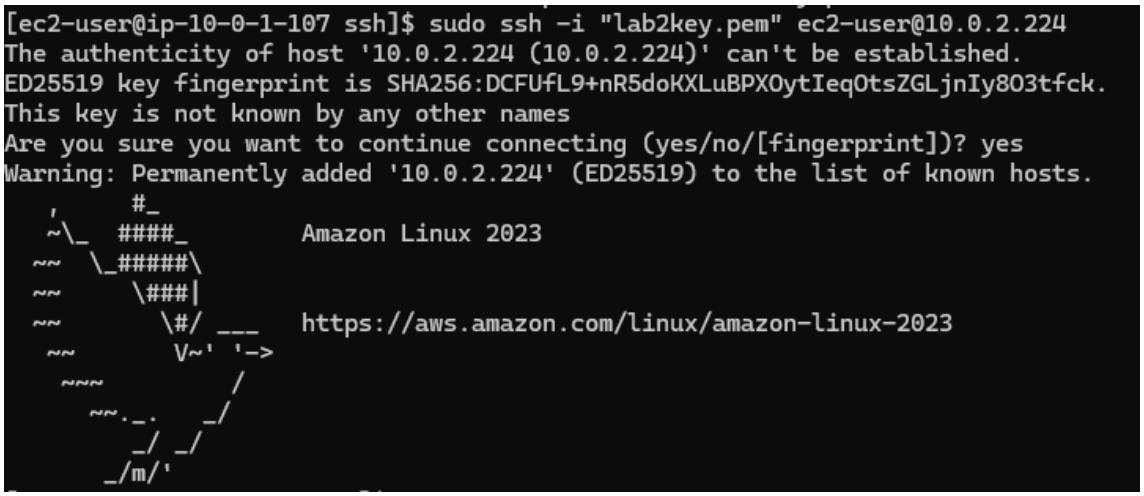
O primeiro teste consistiu em conectar-se via SSH à instância Bastion Host localizada na subnet pública. Como esperado, a conexão foi bem-sucedida, utilizando o par de chaves configurado durante a criação da instância.





Acesso à Instância de Base de Dados a partir do Bastion Host

Estando já conectado na Bastion Host, foi realizado um novo teste de SSH, desta vez direcionado à instância DB localizada na subnet privada. Para isso, foi utilizada a chave privada .pem, previamente transferida para a Bastion Host.



Instalação e Testes do MariaDB na Instância DB

Com o acesso garantido à instância de base de dados, foi realizado o processo de instalação do servidor MariaDB (versão 10.5). Esta etapa valida se a instância possui conectividade com a internet através do NAT Gateway, necessária para o download dos pacotes.

```
[ec2-user@ip-10-0-2-224 ~]$ sudo dnf install mariadb105-server -y
Last metadata expiration check: 0:07:03 ago on Wed Apr  9 14:56:12 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 noarch         3.3.10-1.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
 perl-File-Copy             noarch         2.34-477.amzn2023.0.6    amazonlinux     20 k
 perl-FileHandle            noarch         2.03-477.amzn2023.0.6    amazonlinux     16 k
 perl-Math-BigInt           noarch         1:1.9998.39-2.amzn2023.0.2 amazonlinux   202 k
 perl-Math-BigRat           noarch         0.2614-458.amzn2023.0.2  amazonlinux     39 k
 perl-Math-Complex          noarch         1.59-477.amzn2023.0.6    amazonlinux     47 k
 perl-Sys-Hostname          x86_64         1.23-477.amzn2023.0.6    amazonlinux     18 k
 perl-base                  noarch         2.27-477.amzn2023.0.6    amazonlinux     17 k
Installing weak dependencies:
 mariadb105-backup          x86_64         3:10.5.25-1.amzn2023.0.1 amazonlinux    6.3 M
 mariadb105-cracklib-password-check x86_64 3:10.5.25-1.amzn2023.0.1 amazonlinux    15 k
 mariadb105-gssapi-server   x86_64         3:10.5.25-1.amzn2023.0.1 amazonlinux     17 k
 mariadb105-server-utils    x86_64         3:10.5.25-1.amzn2023.0.1 amazonlinux    216 k

Transaction Summary
================================================================================
Install  22 Packages
```

```
[ec2-user@ip-10-0-2-224 ~]$ sudo systemctl start mariadb
[ec2-user@ip-10-0-2-224 ~]$ sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-10-0-2-224 ~]$ mysql --version
mysql  Ver 15.1 Distrib 10.5.25-MariaDB, for Linux (x86_64) using  EditLine wrapper
```

Considerações Finais dos Testes

Os testes realizados comprovam que:

- A instância Bastion Host está acessível externamente via SSH, conforme configurado na subnet pública;
- A instância DB, apesar de não possuir IP público, é acessível a partir do Bastion Host (dentro da mesma VPC), o que confirma a correta segmentação e segurança da rede;
- A instância DB tem acesso à internet através do NAT Gateway, conforme demonstrado na instalação bem-sucedida do MariaDB;
- As configurações dos Security Groups e ACLs permitiram uma comunicação segura e eficiente entre as instâncias;

Conclusão

Este laboratório permitiu consolidar, na prática, os conhecimentos essenciais sobre a criação e gestão de uma infraestrutura de rede segura e escalável na AWS. Foram aplicados conceitos-chave como a segmentação de rede através de subnets públicas e privadas, a configuração de NAT Gateway para garantir acesso controlado à internet a partir de subnets privadas, e a aplicação rigorosa de políticas de segurança através de Security Groups e Network ACLs.