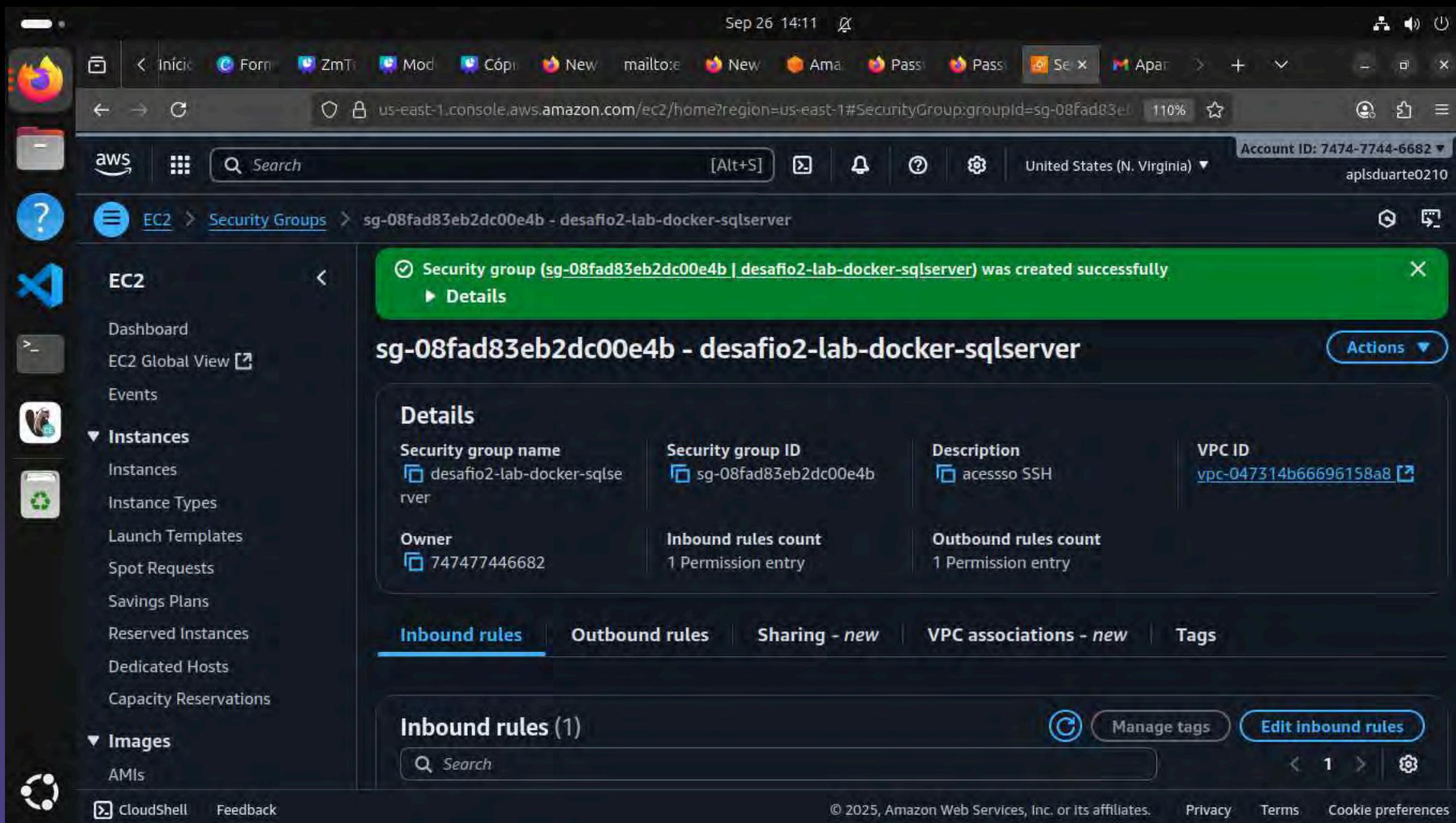


Desafio Labs 2.0

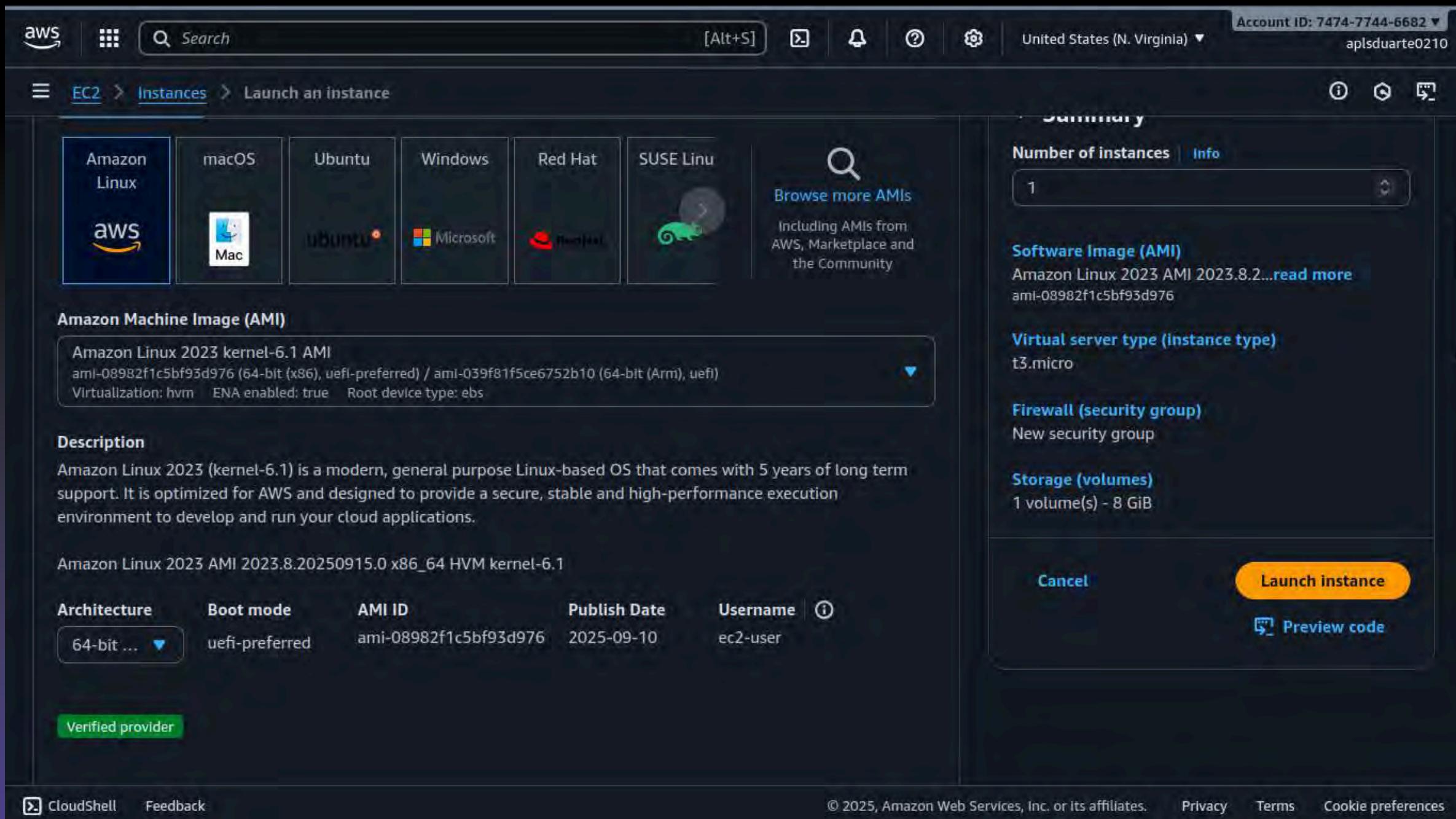
RDS + SQL SERVER

Next Slide

2.Lançamento da instância EC2 - sqlserver.docker

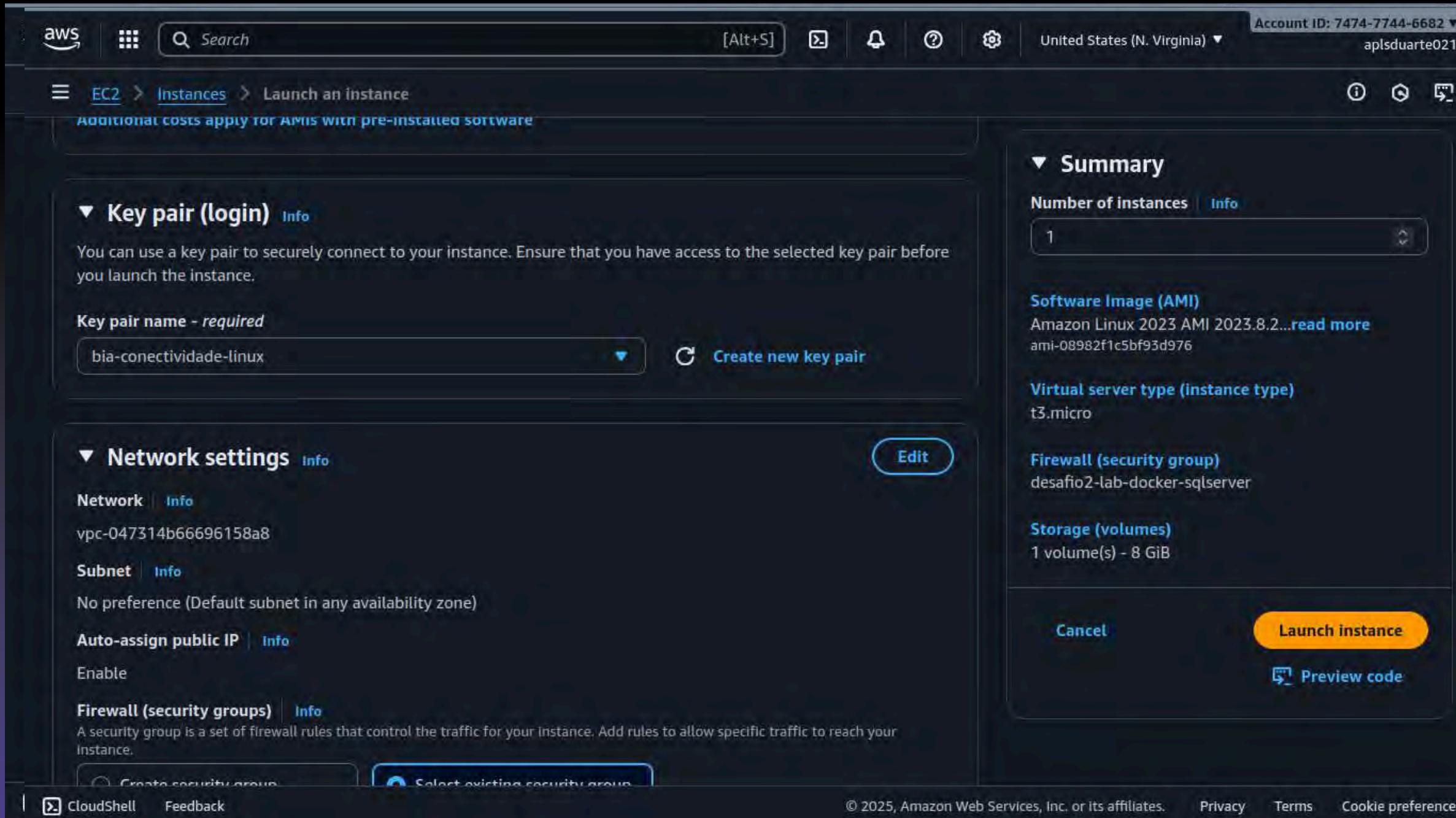


3. Amazon Machine Image (AMI): É a definição do sistema operacional e software inicial da instância. O modelo selecionado é Amazon Linux 2023, que é gratuito no nível Free Tier e otimizado para rodar na AWS.



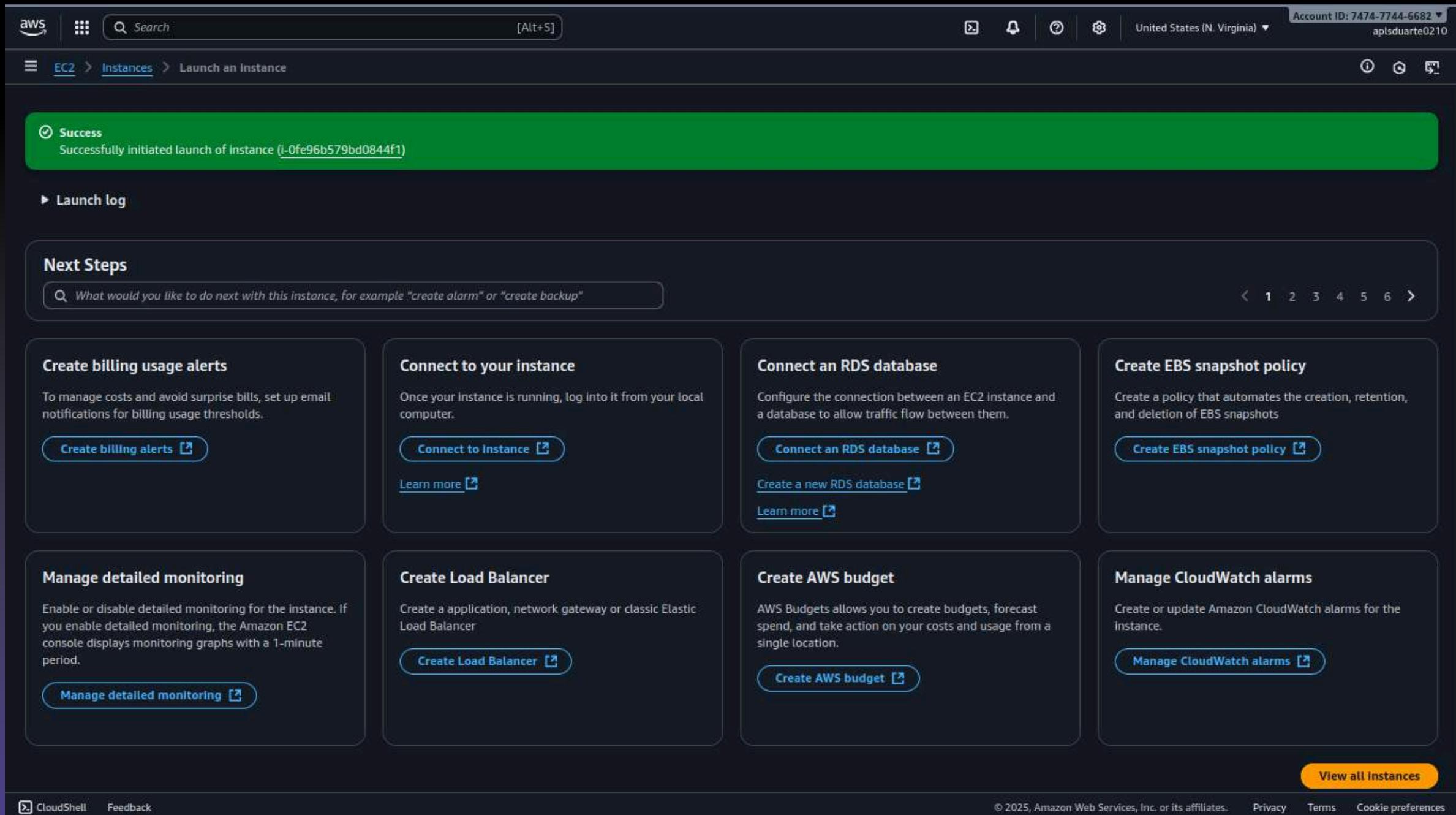
4. Key Pair: Foi criado o par de chaves bia-conectividade para permitir acesso seguro à instância. Utilizamos o algoritmo RSA, por ser amplamente aceito e confiável na AWS. Esse algoritmo garante um bom equilíbrio entre segurança e compatibilidade.

A chave privada é gerada no formato .pem (Privacy Enhanced Mail). O arquivo .pem é essencial para conexões SSH seguras, sem uso de senha.



5. Instância EC2 criada com sucesso na AWS (Região N. Virgínia).

A tela confirma a inicialização bem-sucedida da instância, exibindo o ID 0fe96b579bd0844f1, pronta para acesso remoto e configuração dos próximos recursos do ambiente.



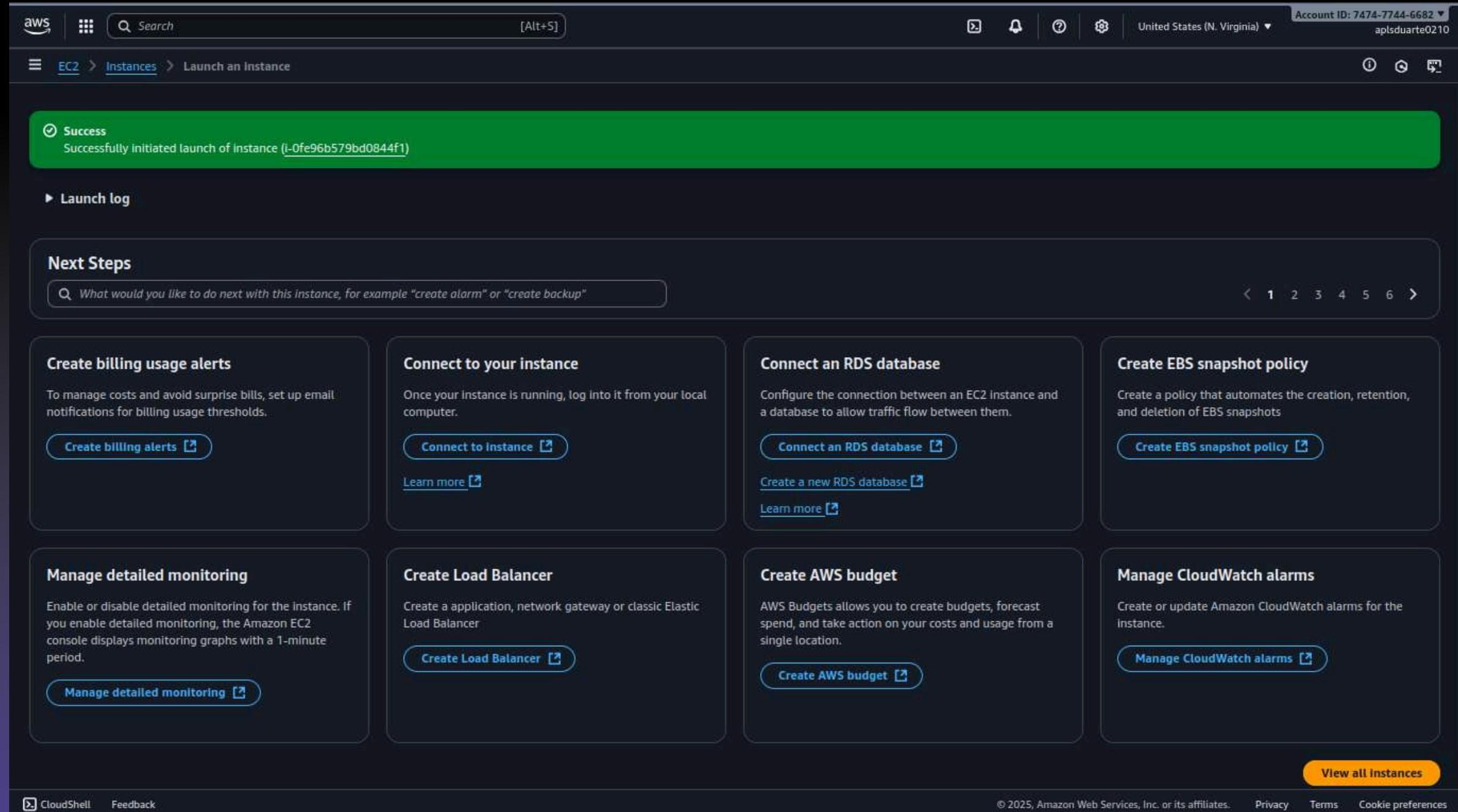
The screenshot shows the AWS EC2 Instances Launch an Instance page. At the top, there is a green success message: "Successfully initiated launch of Instance (i-0fe96b579bd0844f1)". Below this, there is a "Next Steps" section with a search bar and a navigation menu (1-6). The main area contains eight cards:

- Create billing usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. Includes a "Create billing alerts" button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a "Connect to instance" button and a "Learn more" link.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a "Connect an RDS database" button and a "Create a new RDS database" link.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a "Create EBS snapshot policy" button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period. Includes a "Manage detailed monitoring" button.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a "Create Load Balancer" button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Includes a "Create AWS budget" button.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a "Manage CloudWatch alarms" button.

At the bottom right, there is a "View all instances" button. The footer includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

5. Instância EC2 criada com sucesso na AWS (Região N. Virgínia).

A tela confirma a inicialização correta do ambiente, com o ID da instância gerado e pronta para configuração e execução dos próximos passos do laboratório.

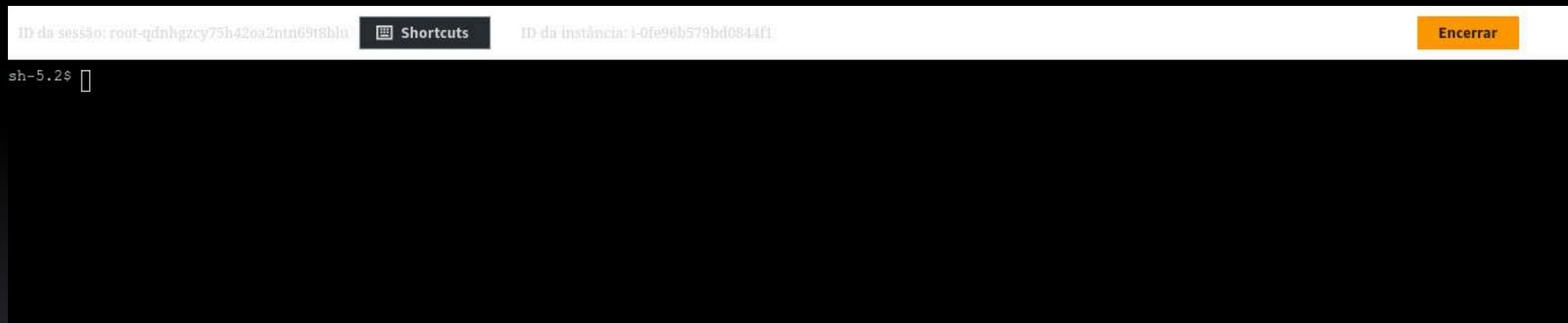


The screenshot shows the AWS EC2 Instances Launch an Instance page. At the top, there is a green success message: "Successfully initiated launch of instance (i-0fe96b579bd0844f1)". Below this, there is a "Next Steps" section with a search bar and a navigation menu (1-6). The main area contains eight cards:

- Create billing usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing usage thresholds. Includes a "Create billing alerts" button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a "Connect to instance" button and a "Learn more" link.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a "Connect an RDS database" button and a "Create a new RDS database" link.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a "Create EBS snapshot policy" button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period. Includes a "Manage detailed monitoring" button.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a "Create Load Balancer" button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Includes a "Create AWS budget" button.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a "Manage CloudWatch alarms" button.

At the bottom right, there is a "View all instances" button. The footer includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

6. Conexão Session Manager - Acesso à EC2 realizada com sucesso!

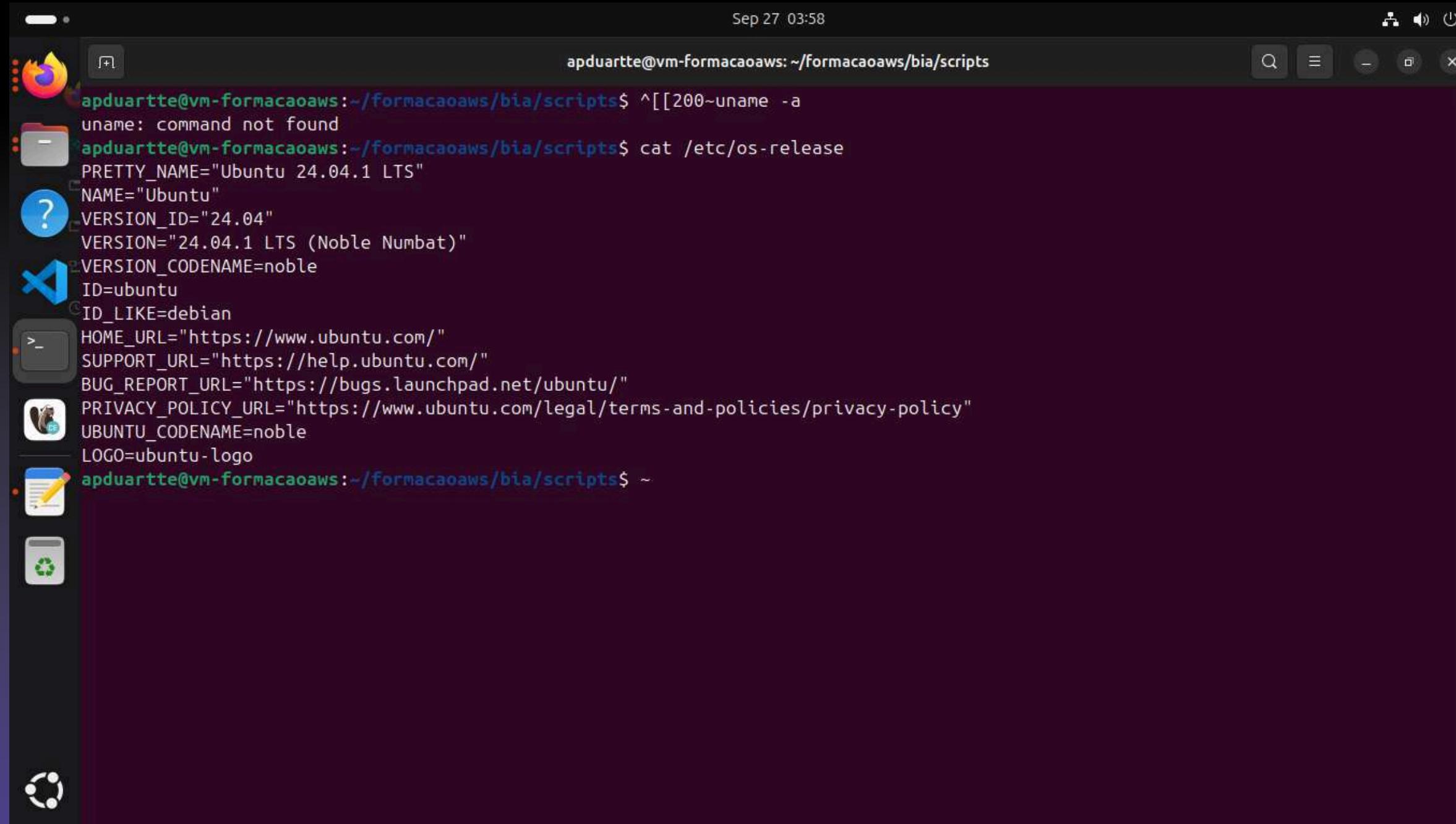


SSH vs Session Manager – Acesso à EC2		
Aspecto	SSH Tradicional	Session Manager
Chave de acesso	Precisa de arquivo <code>.pem</code> (key pair)	Não precisa de chave <code>.pem</code>
Portas	Porta 22/TCP deve estar liberada no Security Group	Não requer porta aberta (usa o agente SSM)
Segurança	Exposição da porta SSH pode ser alvo de ataques	Mais seguro, sem porta exposta
Dependências	Cliente SSH instalado localmente + permissões corretas	Apenas navegador + console AWS
Controle de auditoria	Logs de sessão não são gravados automaticamente	Pode registrar logs no CloudWatch/CloudTrail
Facilidade	Requer configurar IP, chave e permissões	Basta clicar em <code>Conectar</code> no Console AWS
Casos de uso	Bom para ambientes tradicionais com acesso direto	Ideal para ambientes seguros e gerenciados (prod/hardening)

Setembro/2025

apduartte@gmail.com

7. Identificação do sistema operacional concluída com sucesso. A imagem mostra a instância EC2 executando Ubuntu 24.04.1 LTS(Noble Numbat), baseado em Debian, garantindo estabilidade, suporte prolongado e compatibilidade com ferramentas como Docker, Node.js e Python..

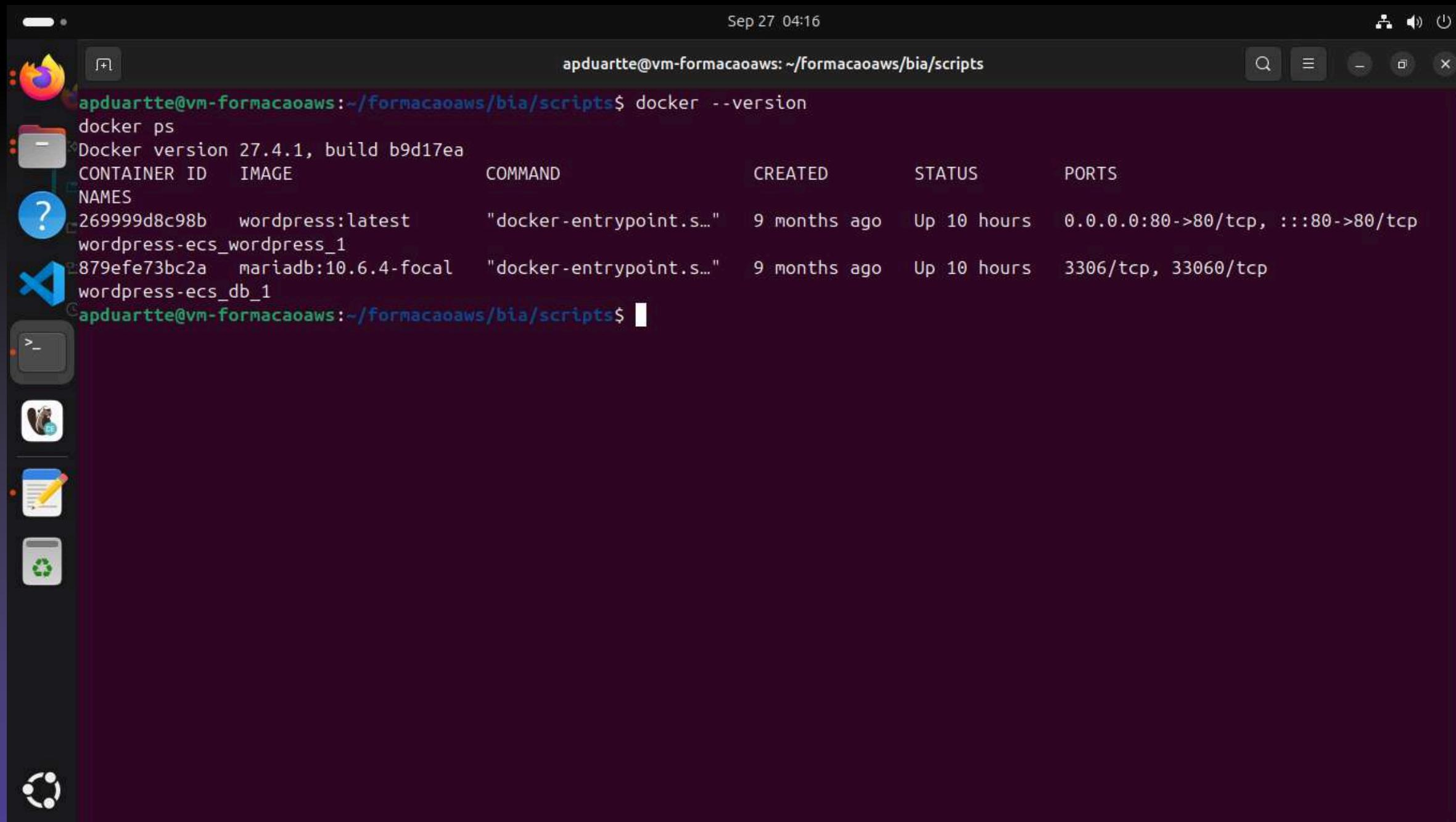


The screenshot shows a terminal window with a dark theme. The title bar reads "apduartte@vm-formacaoaws: ~/formacaoaws/bia/scripts" and the date "Sep 27 03:58". The terminal displays the following command-line session:

```
apduartte@vm-formacaoaws:~/formacaoaws/bia/scripts$ ^[[200~uname -a
uname: command not found
apduartte@vm-formacaoaws:~/formacaoaws/bia/scripts$ cat /etc/os-release
PRETTY_NAME="Ubuntu 24.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.1 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
apduartte@vm-formacaoaws:~/formacaoaws/bia/scripts$ ~
```

08. Verificação do ambiente Linux concluída com sucesso.

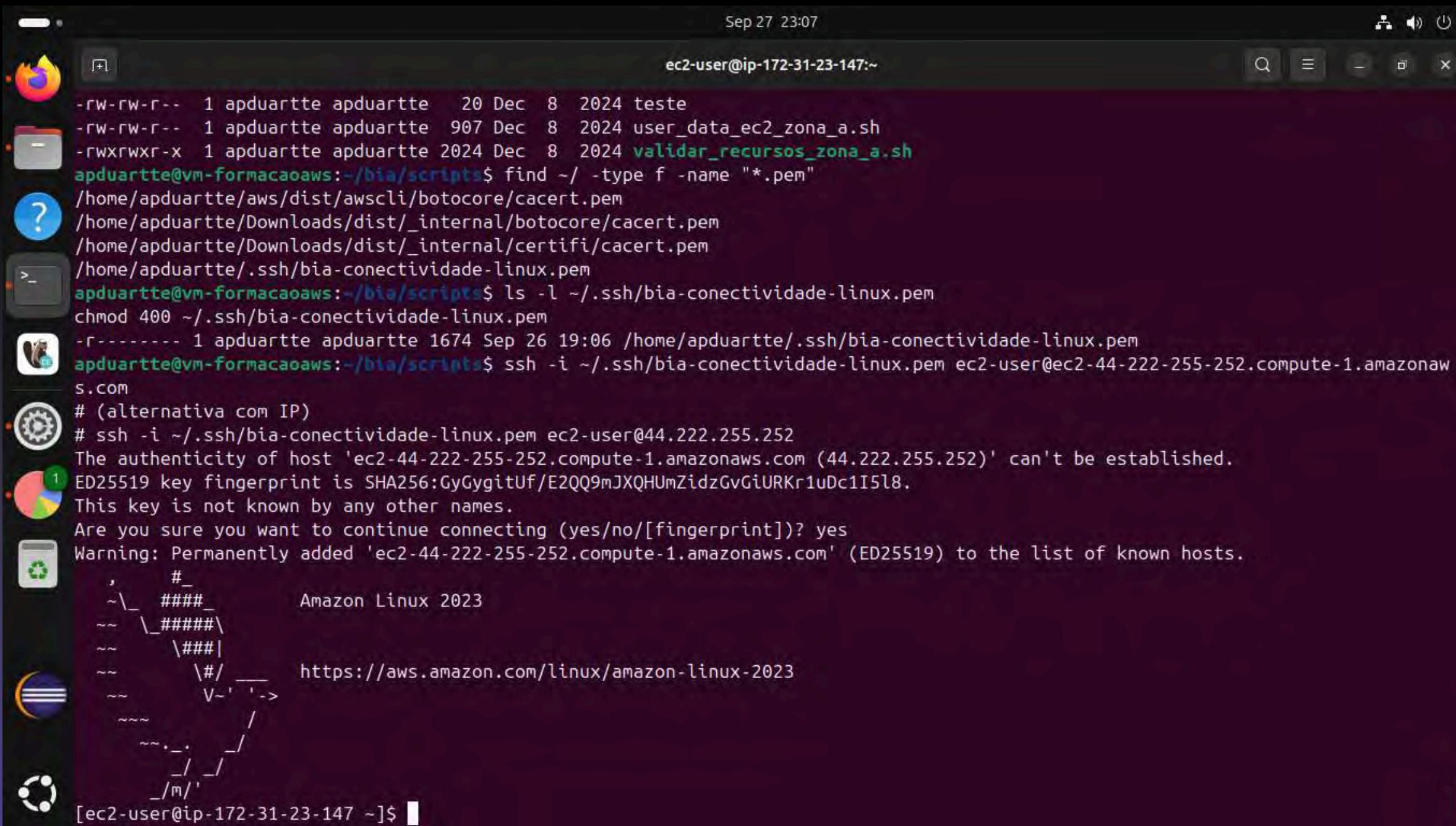
A imagem mostra o terminal confirmando o uso da distro Ubuntu 24.04.1 LTS (Noble Numbat) baseada em Debian, com o Docker versão 27.4.1 ativo e containers em execução, demonstrando ambiente estável e pronto para operação.



The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open, displaying the output of a Docker command. The terminal title bar reads "apduartte@vm-formacaoaws: ~/formacaoaws/bia/scripts". The terminal content shows:

```
apduartte@vm-formacaoaws:~/formacaoaws/bia/scripts$ docker --version
docker ps
Docker version 27.4.1, build b9d17ea
CONTAINER ID   IMAGE          COMMAND           CREATED        STATUS         PORTS
NAMES
269999d8c98b   wordpress:latest "docker-entrypoint.s..." 9 months ago   Up 10 hours   0.0.0.0:80->80/tcp, :::80->80/tcp
wordpress-ecs_wordpress_1
879efe73bc2a   mariadb:10.6.4-focal "docker-entrypoint.s..." 9 months ago   Up 10 hours   3306/tcp, 33060/tcp
wordpress-ecs_db_1
apduartte@vm-formacaoaws:~/formacaoaws/bia/scripts$
```

9. Conexão SSH estabelecida com sucesso na instância EC2 Linux, utilizando a chave bia-conectividade-linux.pem, confirmando o acesso seguro ao ambiente AWS.

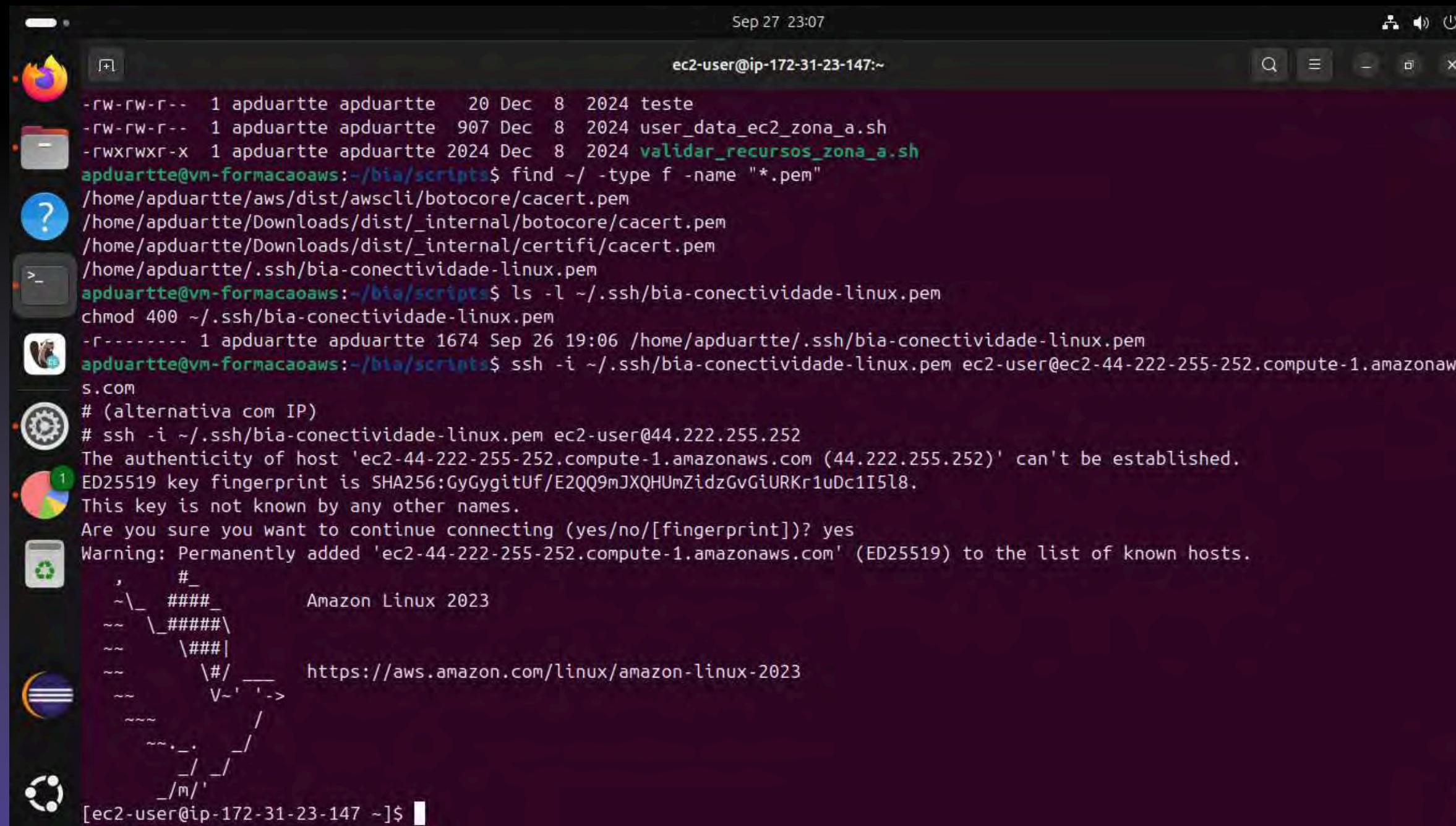


The screenshot shows a terminal window titled "ec2-user@ip-172-31-23-147:~". The terminal displays the following command and its output:

```
apduartte@vm-formacaoaws:~/bia/scripts$ find ~/ -type f -name "*.pem"
/home/apduartte/aws/dist/awscli/botocore/cacert.pem
/home/apduartte/Downloads/dist/_internal/botocore/cacert.pem
/home/apduartte/Downloads/dist/_internal/certifi/cacert.pem
/home/apduartte/.ssh/bia-conectividade-linux.pem
apduartte@vm-formacaoaws:~/bia/scripts$ ls -l ~/.ssh/bia-conectividade-linux.pem
chmod 400 ~/.ssh/bia-conectividade-linux.pem
-r----- 1 apduartte apduartte 1674 Sep 26 19:06 /home/apduartte/.ssh/bia-conectividade-linux.pem
apduartte@vm-formacaoaws:~/bia/scripts$ ssh -i ~/.ssh/bia-conectividade-linux.pem ec2-user@ec2-44-222-255-252.compute-1.amazonaws.com
# (alternativa com IP)
# ssh -i ~/.ssh/bia-conectividade-linux.pem ec2-user@44.222.255.252
The authenticity of host 'ec2-44-222-255-252.compute-1.amazonaws.com (44.222.255.252)' can't be established.
ED25519 key fingerprint is SHA256:GyGygitUF/E2QQ9mJXQHUmZidzGvGiURKr1uDc1I5l8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-222-255-252.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
```

The terminal also shows the system's root file system structure with icons on the left side.

10. Verificação do ambiente Docker realizada com sucesso, confirmando a versão 25.0.8 instalada e pronta para execução dos containers no servidor EC2.



The screenshot shows a terminal window titled "ec2-user@ip-172-31-23-147:~". The terminal displays the following command and its output:

```
apduartte@vm-formacaoaws:~/bia/scripts$ find ~/ -type f -name "*.pem"
/home/apduartte/aws/dist/awscli/botocore/cacert.pem
/home/apduartte/Downloads/dist/_internal/botocore/cacert.pem
/home/apduartte/Downloads/dist/_internal/certifi/cacert.pem
/home/apduartte/.ssh/bia-conectividade-linux.pem
apduartte@vm-formacaoaws:~/bia/scripts$ ls -l ~/.ssh/bia-conectividade-linux.pem
chmod 400 ~/.ssh/bia-conectividade-linux.pem
-r----- 1 apduartte apduartte 1674 Sep 26 19:06 /home/apduartte/.ssh/bia-conectividade-linux.pem
apduartte@vm-formacaoaws:~/bia/scripts$ ssh -i ~/.ssh/bia-conectividade-linux.pem ec2-user@ec2-44-222-255-252.compute-1.amazonaws.com
# (alternativa com IP)
# ssh -i ~/.ssh/bia-conectividade-linux.pem ec2-user@44.222.255.252
The authenticity of host 'ec2-44-222-255-252.compute-1.amazonaws.com (44.222.255.252)' can't be established.
ED25519 key fingerprint is SHA256:GyGygitUF/E2QQ9mJXQHUmZidzGvGiURKr1uDc1I5l8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-222-255-252.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
```

The terminal window has a dark theme with icons on the left side. The status bar at the bottom shows the command prompt "[ec2-user@ip-172-31-23-147 ~]\$".

11.. Backup AdventureWorks2022.bak baixado e instalado com sucesso na instância EC2, confirmando a importação correta do arquivo para o ambiente Docker do SQL Server.

The screenshot shows a terminal window titled "ec2-user@ip-172-31-23-147:~". The terminal output is as follows:

```
Sep 28 00:20
HTTP request sent, awaiting response... 302 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/53698446/bea81028-d987-42dd-972d-9417c5e6bd1c?sp=r&sv=2018-11-09&sr=b&spr=https&se=2025-09-28T01%3A17%3A26Z&rscd=attachment%3B+filename%3DAdventureWorks2022.bak&rsct=application%2Foctet-stream&skoid=96c2d410-5711-43a1-aedd-ab1947aa7ab0&sktid=398a6654-997b-47e9-b12b-9515b896b4de&skt=2025-09-28T00%3A16%3A54Z&ske=2025-09-28T01%3A17%3A26Z&sks=b&skv=2018-11-09&sig=m%2FKjJy0UVVG8iMg%2FvF8QVuFGORqqcoJe%2BZQqKC88L1Y%3D&jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkijoicmVsZWfzs1hc3NldHMuZ2l0aHvidXNlcNmVbnRlbnQuY29tIiwi2V5Ijoia2V5MSIsImV4cCI6MTc10TAxOTEwNSibmJmIjoxNzU5MDE4ODA1LCJwYXRoijoicmVsZWfzZWfzc2V0cHJvZHvjdGlvbisibG9iLmNvcmUud2luZG93cy5uZXQifQ.ocLo0nTkk7E6Xa86Hx2t66Ma0V6_iz-seprbwTAGKp0&response-content-disposition=attachment%3B%20filename%3DAdventureWorks2022.bak&response-content-type=application%2Foctet-stream [following]
--2025-09-28 00:20:05-- https://release-assets.githubusercontent.com/github-production-release-asset/53698446/bea81028-d987-42dd-972d-9417c5e6bd1c?sp=r&sv=2018-11-09&sr=b&spr=https&se=2025-09-28T01%3A17%3A26Z&rscd=attachment%3B+filename%3DAdventureWorks2022.bak&rsct=application%2Foctet-stream&skoid=96c2d410-5711-43a1-aedd-ab1947aa7ab0&sktid=398a6654-997b-47e9-b12b-9515b896b4de&skt=2025-09-28T00%3A16%3A54Z&ske=2025-09-28T01%3A17%3A26Z&sks=b&skv=2018-11-09&sig=m%2FKjJy0UVVG8iMg%2FvF8QVuFGORqqcoJe%2BZQqKC88L1Y%3D&jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkijoicmVsZWfzs1hc3NldHMuZ2l0aHvidXNlcNmVbnRlbnQuY29tIiwi2V5Ijoia2V5MSIsImV4cCI6MTc10TAxOTEwNSibmJmIjoxNzU5MDE4ODA1LCJwYXRoijoicmVsZWfzZWfzc2V0cHJvZHvjdGlvbisibG9iLmNvcmUud2luZG93cy5uZXQifQ.ocLo0nTkk7E6Xa86Hx2t66Ma0V6_iz-seprbwTAGKp0&response-content-disposition=attachment%3B%20filename%3DAdventureWorks2022.bak&response-content-type=application%2Foctet-stream
Resolving release-assets.githubusercontent.com (release-assets.githubusercontent.com)... 185.199.109.133, 185.199.111.133, 185.199.110.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 209838080 (200M) [application/octet-stream]
Saving to: 'AdventureWorks2022.bak'

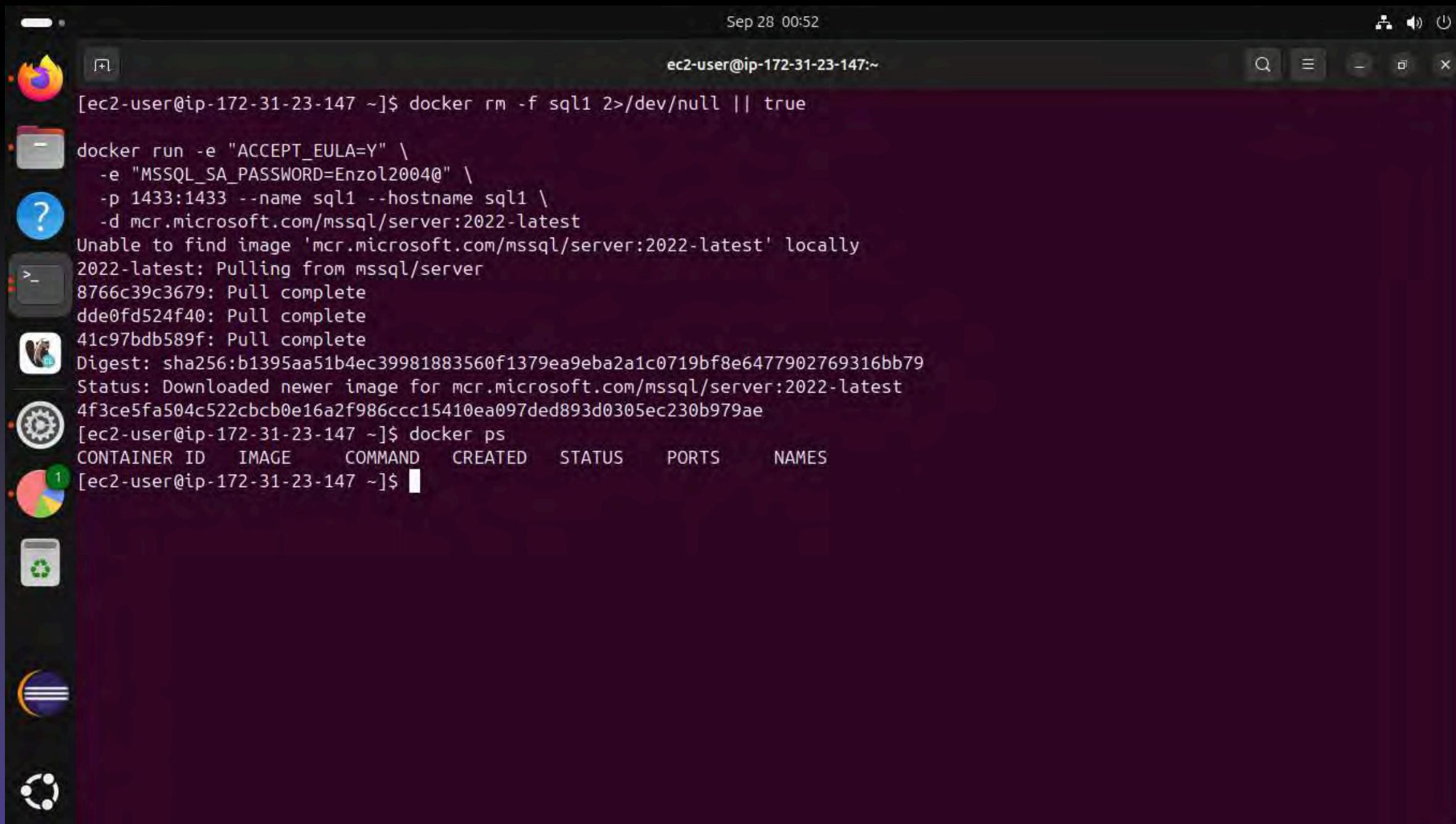
AdventureWorks2022.bak          100%[=====] 200.12M  91.7MB/s  in 2.2s
2025-09-28 00:20:07 (91.7 MB/s) - 'AdventureWorks2022.bak' saved [209838080/209838080]

[ec2-user@ip-172-31-23-147 ~]$ ls
AdventureWorks2022.bak
[ec2-user@ip-172-31-23-147 ~]$
```

The screenshot shows a terminal window titled "ec2-user@ip-172-31-23-147:~". The terminal output is as follows:

```
Sep 28 00:52
[ec2-user@ip-172-31-23-147 ~]$ docker rm -f sql1 2>/dev/null || true
[ec2-user@ip-172-31-23-147 ~]$ docker run -e "ACCEPT_EULA=Y" \
-e "MSSQL_SA_PASSWORD=Enzol2004@" \
-p 1433:1433 --name sql1 --hostname sql1 \
-d mcr.microsoft.com/mssql/server:2022-latest
Unable to find image 'mcr.microsoft.com/mssql/server:2022-latest' locally
2022-latest: Pulling from mssql/server
8766c39c3679: Pull complete
dde0fd524f40: Pull complete
41c97bdb589f: Pull complete
Digest: sha256:b1395aa51b4ec39981883560f1379ea9eba2a1c0719bf8e6477902769316bb79
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2022-latest
4f3ce5fa504c522cbcb0e16a2f986ccc15410ea097ded893d0305ec230b979ae
[ec2-user@ip-172-31-23-147 ~]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[ec2-user@ip-172-31-23-147 ~]$
```

13. Coloquei o SQL Server 2022 para rodar em uma EC2 usando Docker e conectei com segurança via AWS Systems Manager (port forwarding) — sem expor a porta 1433 na internet. Ajustei permissões (UID 10001), validei a escuta com ss -lntp e conectei no DBeaver via localhost:1436. Resultado: ambiente reproduzível, seguro e pronto para restaurar o AdventureWorks e praticar consultas. 🚀



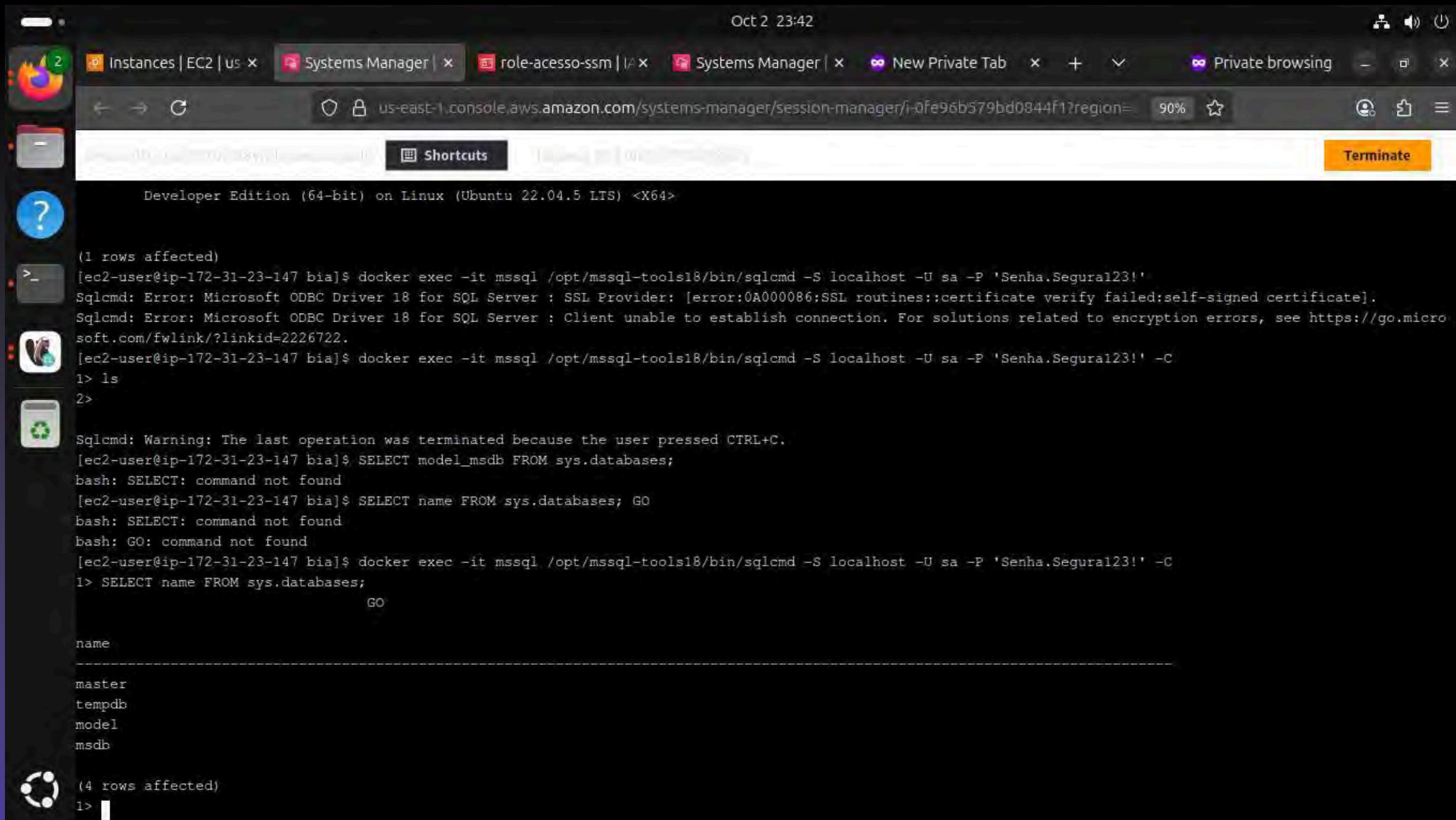
The screenshot shows a terminal window on a Linux desktop environment. The terminal output is as follows:

```
Sep 28 00:52
[ec2-user@ip-172-31-23-147 ~]$ docker rm -f sql1 2>/dev/null || true
[ec2-user@ip-172-31-23-147 ~]$ docker run -e "ACCEPT_EULA=Y" \
-e "MSSQL_SA_PASSWORD=Enzol2004@" \
-p 1433:1433 --name sql1 --hostname sql1 \
-d mcr.microsoft.com/mssql/server:2022-latest
Unable to find image 'mcr.microsoft.com/mssql/server:2022-latest' locally
2022-latest: Pulling from mssql/server
8766c39c3679: Pull complete
dde0fd524f40: Pull complete
41c97bdb589f: Pull complete
Digest: sha256:b1395aa51b4ec39981883560f1379ea9eba2a1c0719bf8e6477902769316bb79
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2022-latest
4f3ce5fa504c522cbc0e16a2f986ccc15410ea097ded893d0305ec230b979ae
[ec2-user@ip-172-31-23-147 ~]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[ec2-user@ip-172-31-23-147 ~]$
```

14. Conexão SQL Server no EC2 (Docker)

O container do SQL Server 2022 foi executado no EC2 via Docker.

A conexão pelo sqlcmd retornou os bancos padrão: master, tempdb, model e msdb, confirmando funcionamento correto do servidor.



The screenshot shows a Firefox browser window with multiple tabs open, including 'Instances | EC2 | us' and 'Systems Manager'. The main content area is a terminal window titled 'Developer Edition (64-bit) on Linux (Ubuntu 22.04.5 LTS) <X64>'. The terminal output is as follows:

```
(1 rows affected)
[ec2-user@ip-172-31-23-147 bia]$ docker exec -it mssql /opt/mssql-tools18/bin/sqlcmd -S localhost -U sa -P 'Senha.Segura123!'
Sqlcmd: Error: Microsoft ODBC Driver 18 for SQL Server : SSL Provider: [error:0A000086:SSL routines::certificate verify failed:self-signed certificate].
Sqlcmd: Error: Microsoft ODBC Driver 18 for SQL Server : Client unable to establish connection. For solutions related to encryption errors, see https://go.microsoft.com/fwlink/?linkid=2226722.
[ec2-user@ip-172-31-23-147 bia]$ docker exec -it mssql /opt/mssql-tools18/bin/sqlcmd -S localhost -U sa -P 'Senha.Segura123!' -C
1> ls
2>
Sqlcmd: Warning: The last operation was terminated because the user pressed CTRL+C.
[ec2-user@ip-172-31-23-147 bia]$ SELECT model_msdb FROM sys.databases;
bash: SELECT: command not found
[ec2-user@ip-172-31-23-147 bia]$ SELECT name FROM sys.databases; GO
bash: SELECT: command not found
bash: GO: command not found
[ec2-user@ip-172-31-23-147 bia]$ docker exec -it mssql /opt/mssql-tools18/bin/sqlcmd -S localhost -U sa -P 'Senha.Segura123!' -C
1> SELECT name FROM sys.databases;
GO
name
-----
master
tempdb
model
msdb
(4 rows affected)
1>
```

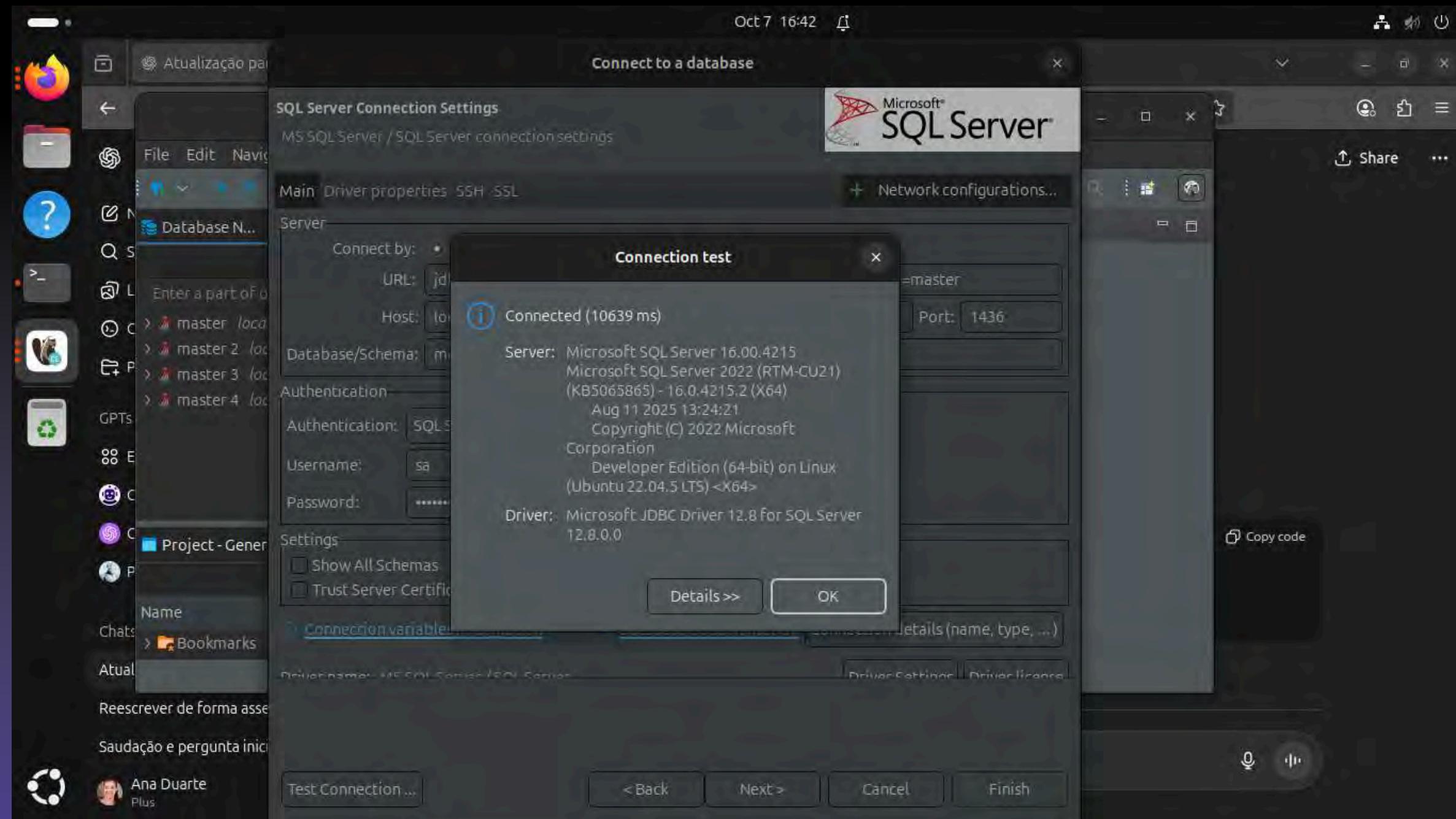
15. Restauração do Banco AdventureWorks2019

A restauração foi realizada a partir do backup em /var/opt/mssql/backup/.

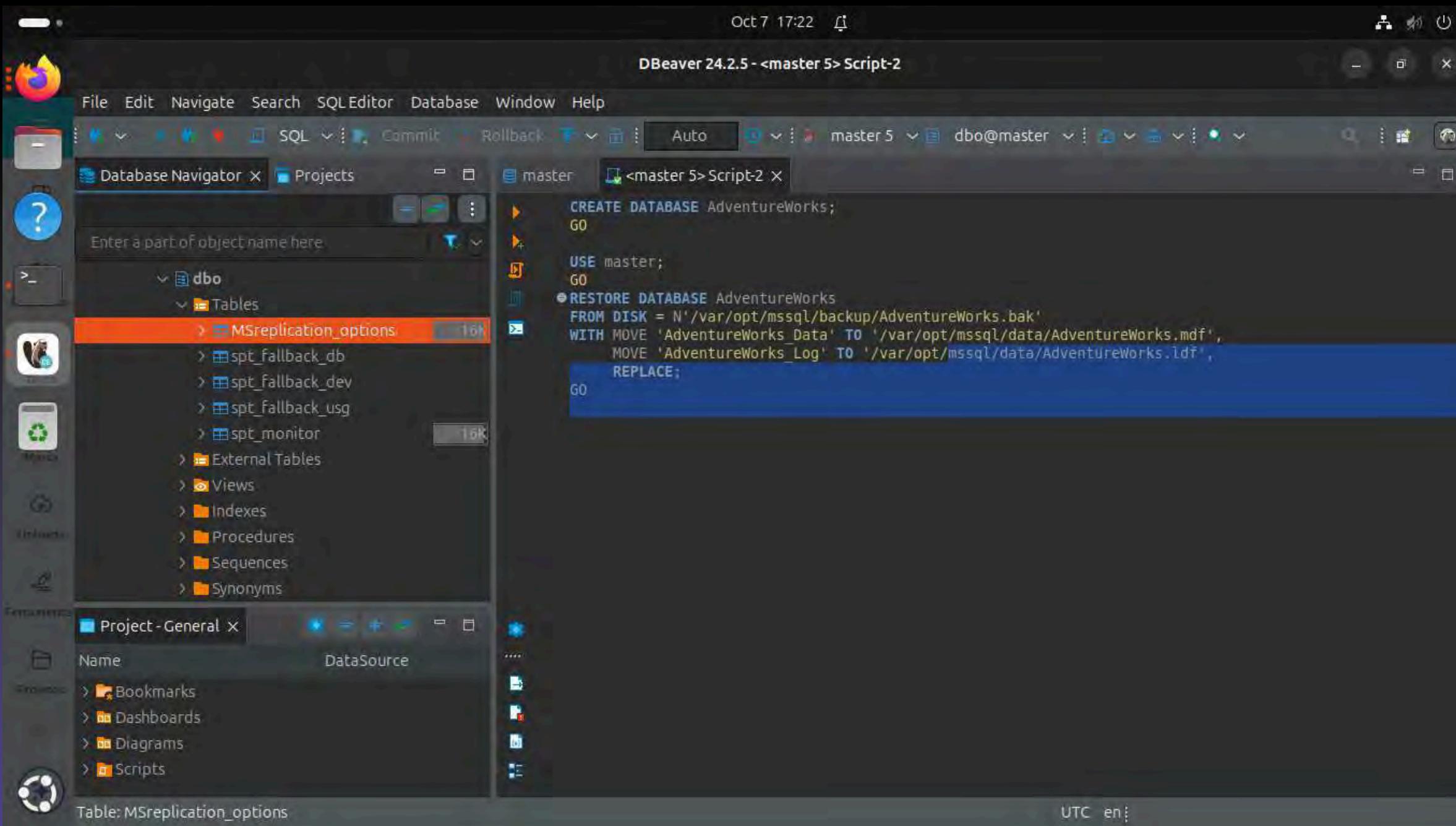
Os arquivos lógicos foram identificados com RESTORE FILELISTONLY e restaurados corretamente com WITH MOVE para o diretório /var/opt/mssql/data.

17. A conexão com o SQL Server 2022 foi realizada com sucesso na porta 1436, validando o mapeamento correto entre o host (Ubuntu/VM) e o container Docker.

O teste confirmou a comunicação ativa e segura entre o cliente DBeaver e o servidor SQL dentro do ambiente Docker, garantindo acesso total às bases e permitindo a execução de consultas.



18. Restauração do banco AdventureWorks realizada com sucesso via DBeaver, utilizando o backup .bak no container Docker e confirmando a conexão ativa na porta 1436.



The screenshot shows the DBeaver interface with the following details:

- Title Bar:** DBeaver 24.2.5 - <master 5> Script-2
- Toolbar:** Includes icons for File, Edit, Navigate, Search, SQL Editor, Database, Window, Help, and various connection and script execution buttons.
- Database Navigator:** Shows the database structure under the master database, including the dbo schema which contains tables like MSreplication_options, spt_fallback_db, spt_fallback_dev, spt_fallback_usg, spt_monitor, External Tables, Views, Indexes, Procedures, Sequences, and Synonyms.
- Script Editor:** Displays the T-SQL script for restoring the AdventureWorks database:

```
CREATE DATABASE AdventureWorks;
GO
USE master;
GO
RESTORE DATABASE AdventureWorks
FROM DISK = N'/var/opt/mssql/backup/AdventureWorks.bak'
WITH MOVE 'AdventureWorks_Data' TO '/var/opt/mssql/data/AdventureWorks.mdf',
MOVE 'AdventureWorks_Log' TO '/var/opt/mssql/data/AdventureWorks.ldf',
REPLACE;
GO
```
- Project Explorer:** Shows a project named "Project - General" with Bookmarks, Dashboards, Diagrams, and Scripts.
- Status Bar:** Shows the current time as Oct 7 17:22, the connection status as master 5, the user as dbo@master, and the language as en.

19. Visualização das tabelas do banco master via DBeaver, confirmando conexão ativa com o SQL Server 2022 (porta 1436) e acesso ao schema dbo.

