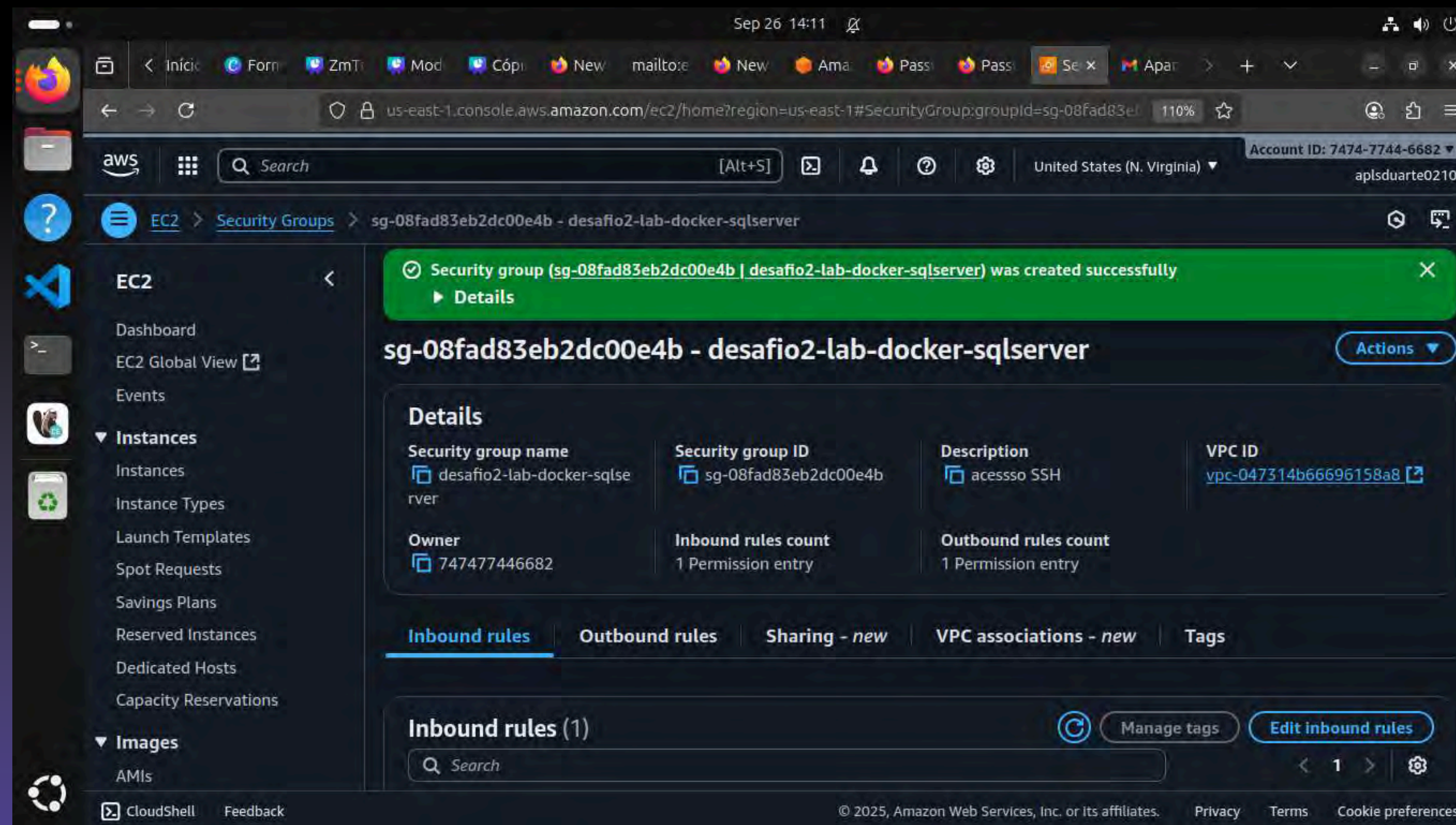


Desafio Labs 2.0

RDS + SQL SERVER

Next Slide

2.Lançamento da instância EC2 - sqlserver.docker

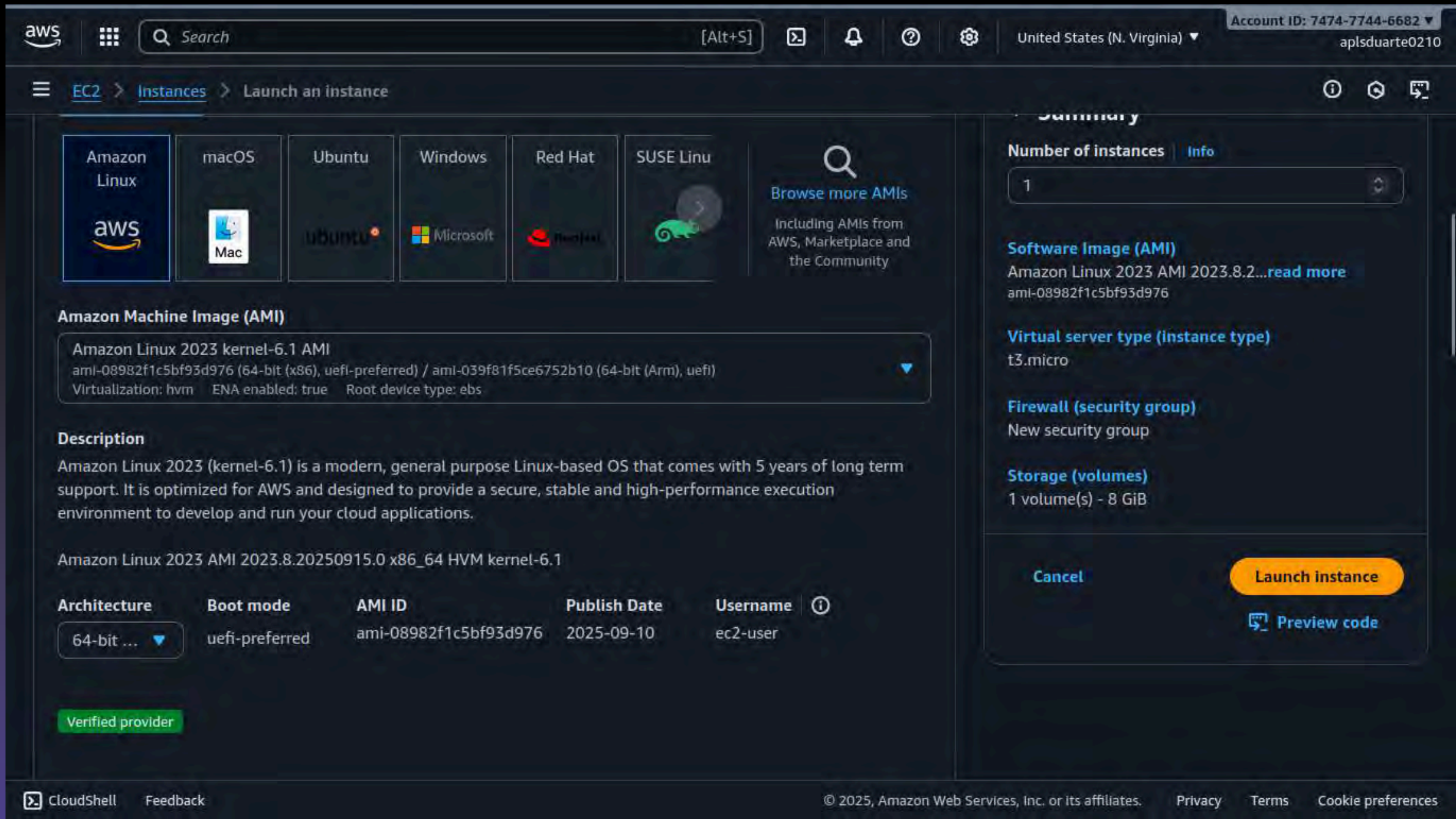


Setembro/2025

-

apduarte@gmail.com

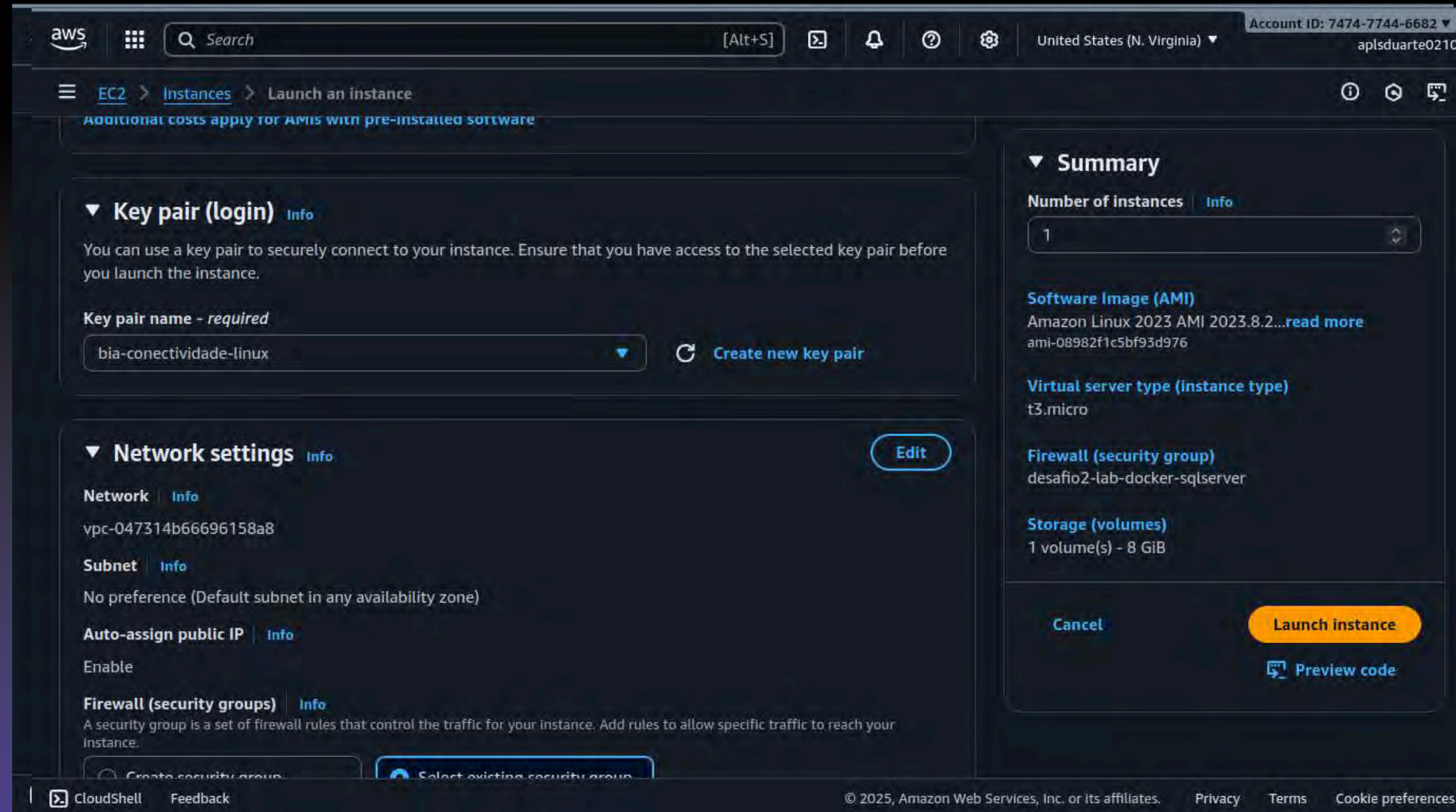
3. Amazon Machine Image (AMI): É a definição do sistema operacional e software inicial da instância. O modelo selecionado é Amazon Linux 2023, que é gratuito no nível Free Tier e otimizado para rodar na AWS.



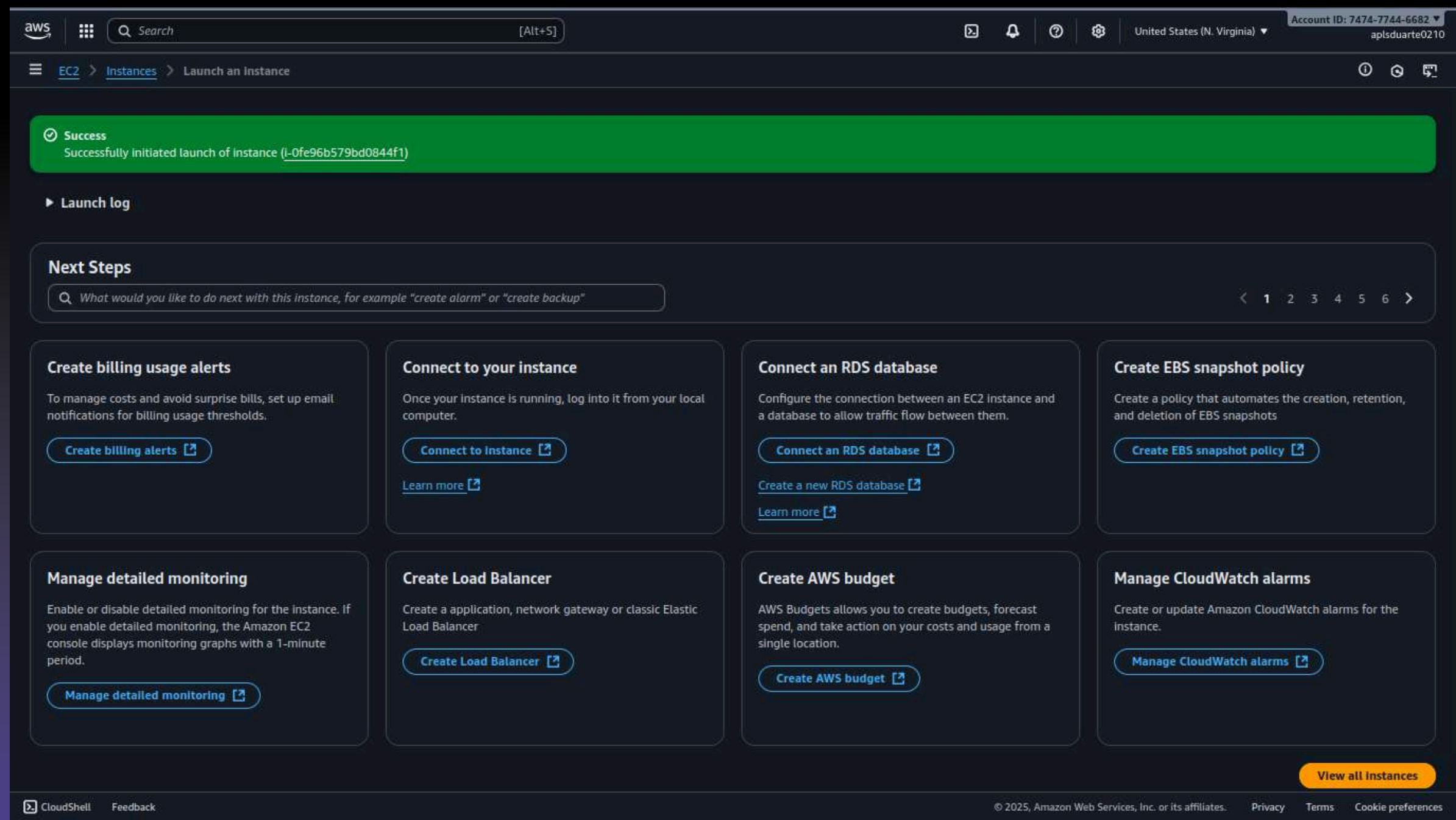
4. Key Pair: Foi criado o par de chaves bia-conectividade para permitir acesso seguro à instância. Utilizamos o algoritmo RSA, por ser amplamente aceito e confiável na AWS. Esse algoritmo garante um bom equilíbrio entre segurança e compatibilidade.

A chave privada é gerada no formato .pem (Privacy Enhanced Mail).

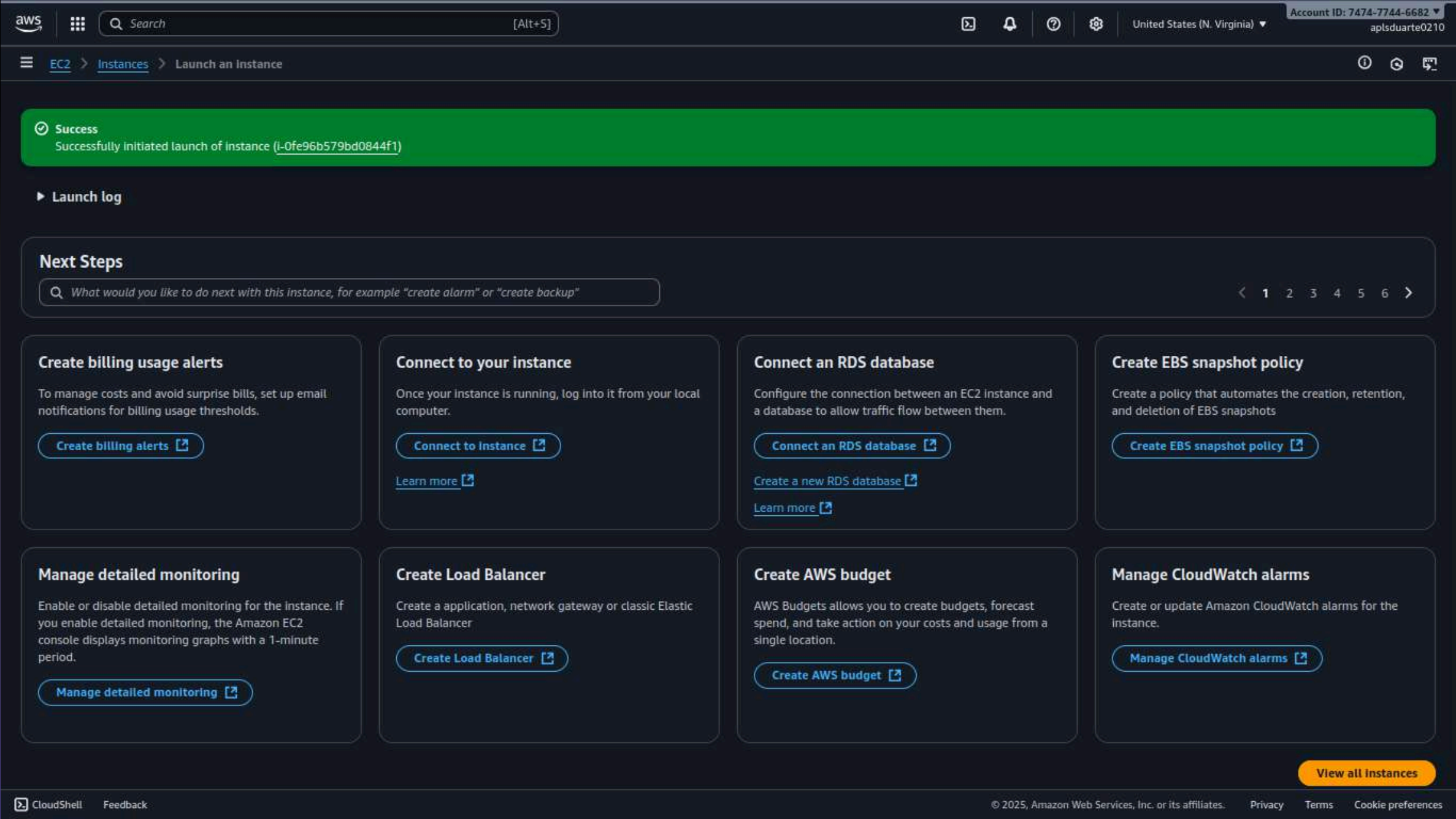
O arquivo .pem é essencial para conexões SSH seguras, sem uso de senha.



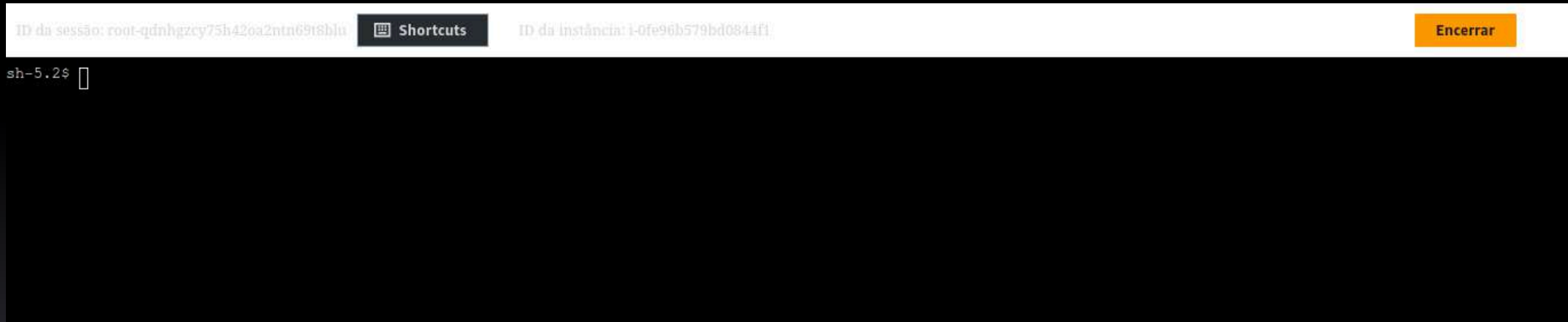
5.Instância EC2 criada com sucesso na AWS (Região N. Virgínia).
A tela confirma a inicialização bem-sucedida da instância, exibindo o ID 0fe96b579bd0844f1, pronta para acesso remoto e configuração dos próximos recursos do ambiente.



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.

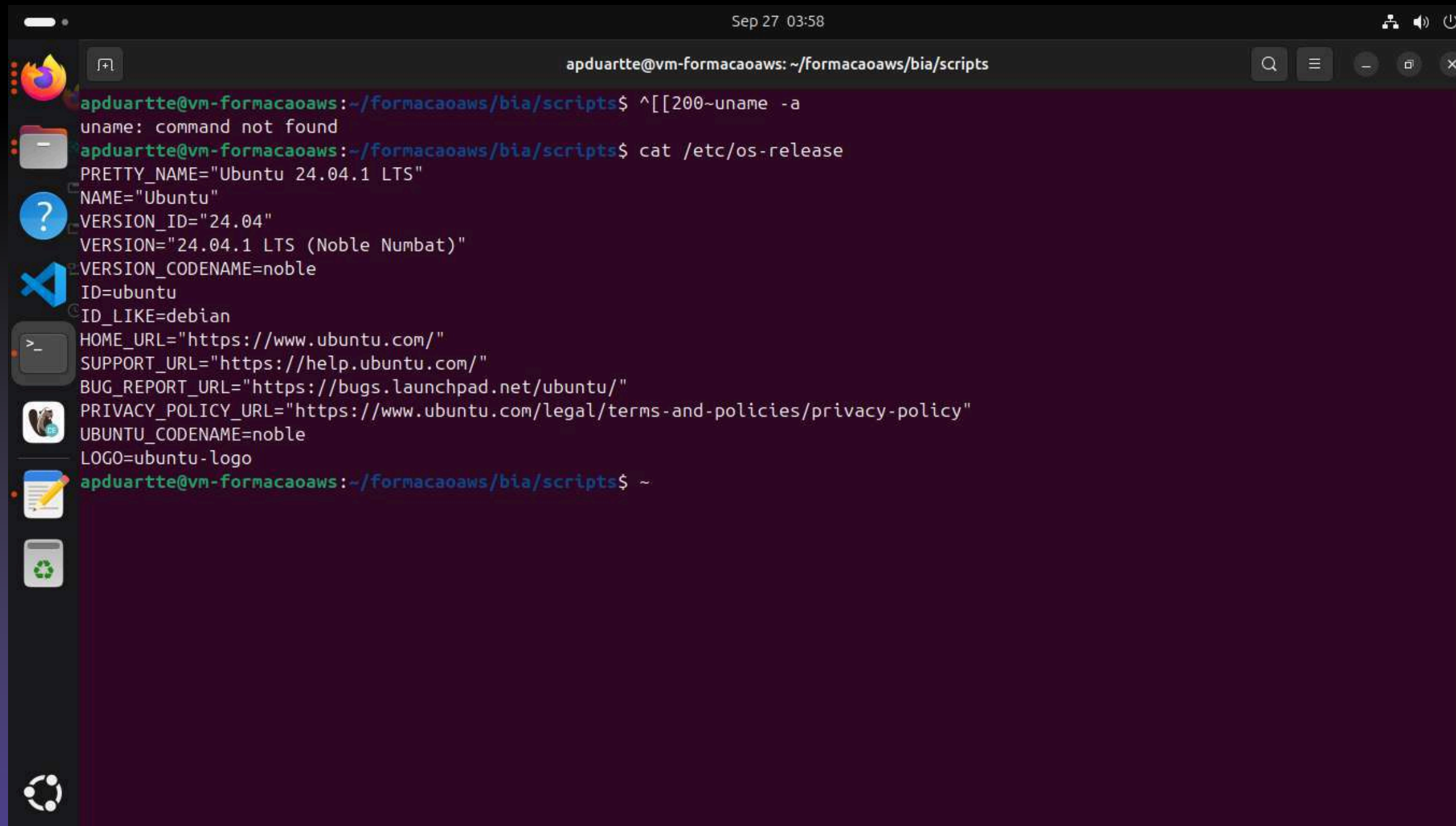


6. Conexão Session Manager - Acesso à EC2 realizada com sucesso!



🔒 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 <i>Conectar</i> no Console AWS
Casos de uso	Bom para ambientes tradicionais com acesso direto	Ideal para ambientes seguros e gerenciados (prod/hardening)

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..

A terminal window titled 'apduartte@vm-formacaoaws: ~/formacaoaws/bia/scripts' with a timestamp 'Sep 27 03:58'. The terminal shows the user 'apduartte' at 'vm-formacaoaws' in the directory '~/formacaoaws/bia/scripts'. The user enters '^[[200~uname -a', which results in 'uname: command not found'. Then, the user enters 'cat /etc/os-release', which outputs the following system information: 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, and LOGO=ubuntu-logo. The terminal ends with the prompt 'apduartte@vm-formacaoaws: ~/formacaoaws/bia/scripts\$ ~'.

```
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.

Sep 27 04:16

apduartte@vm-formacaoaws: ~/formacaoaws/bia/scripts

```
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 `linux.pem`, confirmando o acesso seguro ao ambiente AWS.

```
Sep 27 23:07
```

```
ec2-user@ip-172-31-23-147::~~  
-rw-rw-r-- 1 apduartte apduartte 20 Dec 8 2024 teste  
-rw-rw-r-- 1 apduartte apduartte 907 Dec 8 2024 user_data_ec2_zona_a.sh  
-rwxrwxr-x 1 apduartte apduartte 2024 Dec 8 2024 validar_recursos_zona_a.sh  
apduarte@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/certi/cacert.pem  
/home/apduartte/.ssh/bia-conectividade-linux.pem  
apduarte@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  
apduarte@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/E2QQ9mJXQHUmZidzGvGiURKrIuDc1I5L8.  
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.
```

```
#_ _ _ _ _ Amazon Linux 2023  
~\_\_\_\_\_\_  
~~\_#####\  
~~ \###|  
~~ \#/ https://aws.amazon.com/linux/amazon-linux-2023  
~~ V~'|.->  
~~~~ ~~~~~  
~-._./_____/_____  
_-/_/_/_/_/_/  
-/m/'
```

```
[ec2-user@ip-172-31-23-147 ~]$
```


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

```

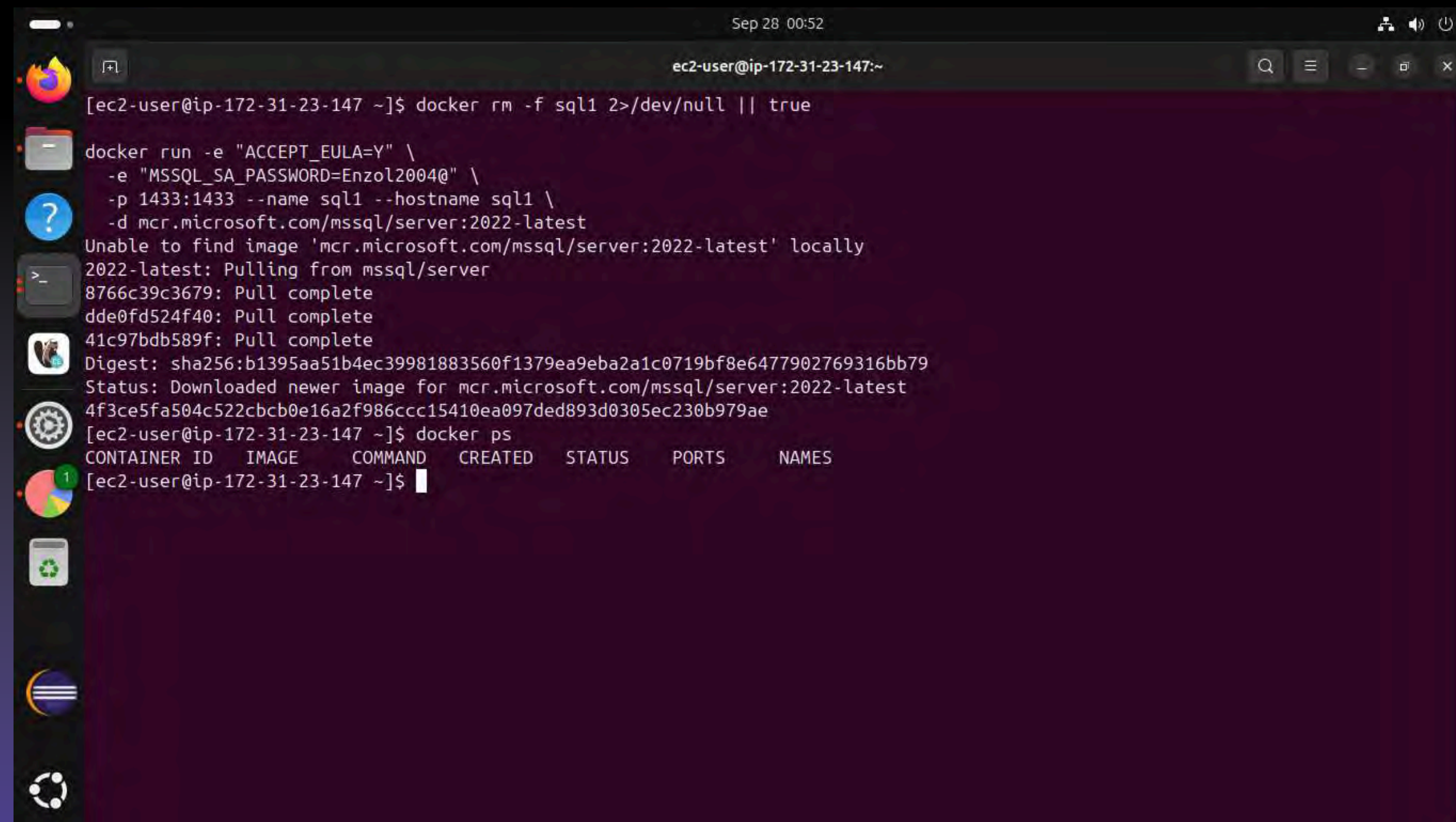
Sep 27 23:07
ec2-user@ip-172-31-23-147:~
-rw-rw-r-- 1 apduartte apduartte 20 Dec 8 2024 teste
-rw-rw-r-- 1 apduartte apduartte 907 Dec 8 2024 user_data_ec2_zona_a.sh
-rwxrwxr-x 1 apduartte apduartte 2024 Dec 8 2024 validar_recursos_zona_a.sh
apduartte@vm-formacaoaws:~/bia/scripts$ find ~/ -type f -name "*.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
-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.
#_
~/_ ##### Amazon Linux 2023
~~ \_#####\
~~ \###|
~~ \#/ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '->
~~~
~~~.~.~
~~/_/_/_/
~~/_m/'
[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.

[illegible]

12. Imagem do SQL Server 2022 baixada com sucesso e container em execução no EC2 via Docker, configurado para escutar na porta 1433



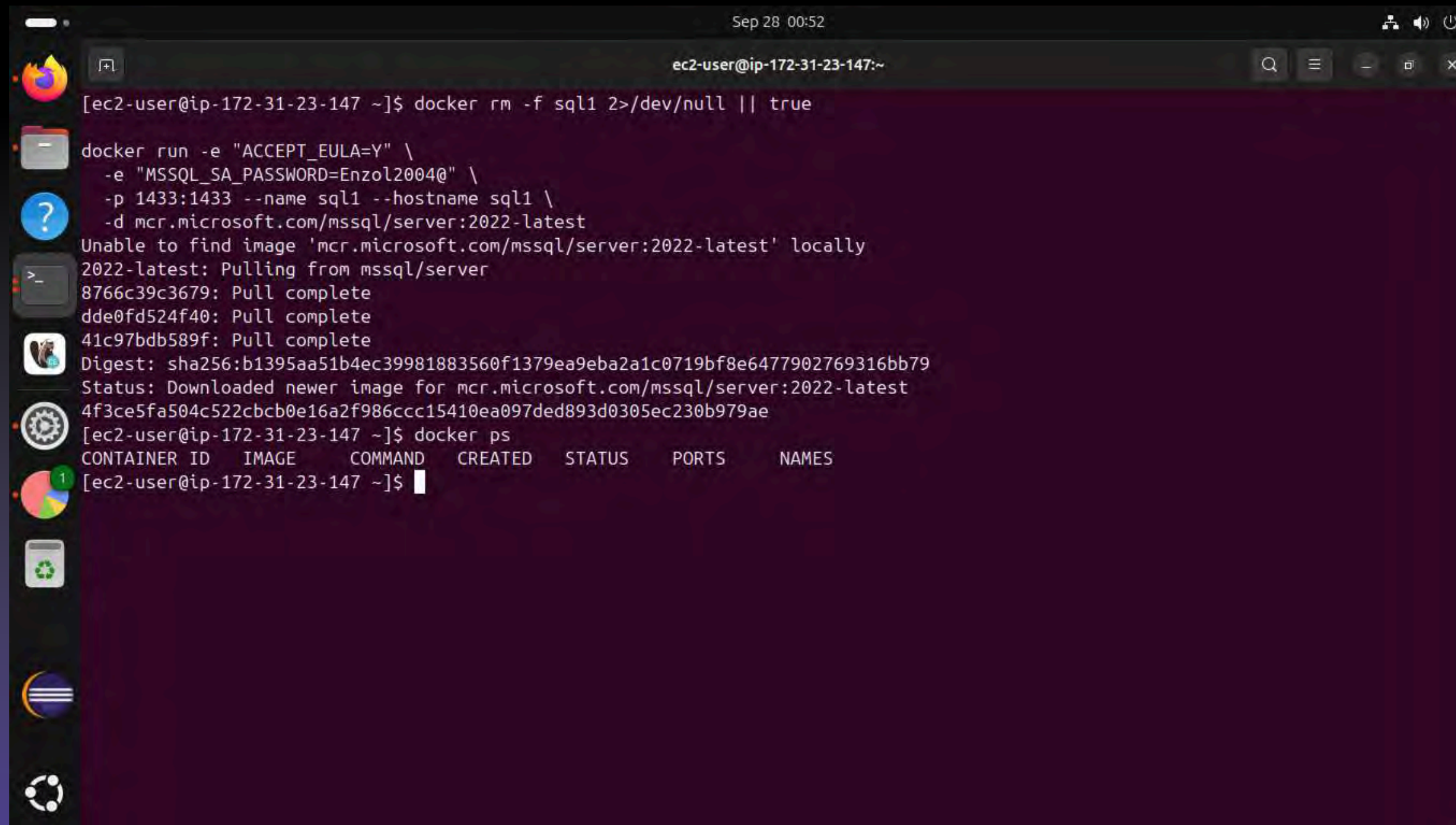
The screenshot shows a terminal window titled "ec2-user@ip-172-31-23-147:~" with a dark background. The terminal displays the following commands and output:

```
[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=Enz0l2004@" \
-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 ~]$
```

The terminal window includes a sidebar with various application icons (Firefox, Files, Nautilus, etc.) and a top bar showing the date and time as "Sep 28 00:52".

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. 🚀



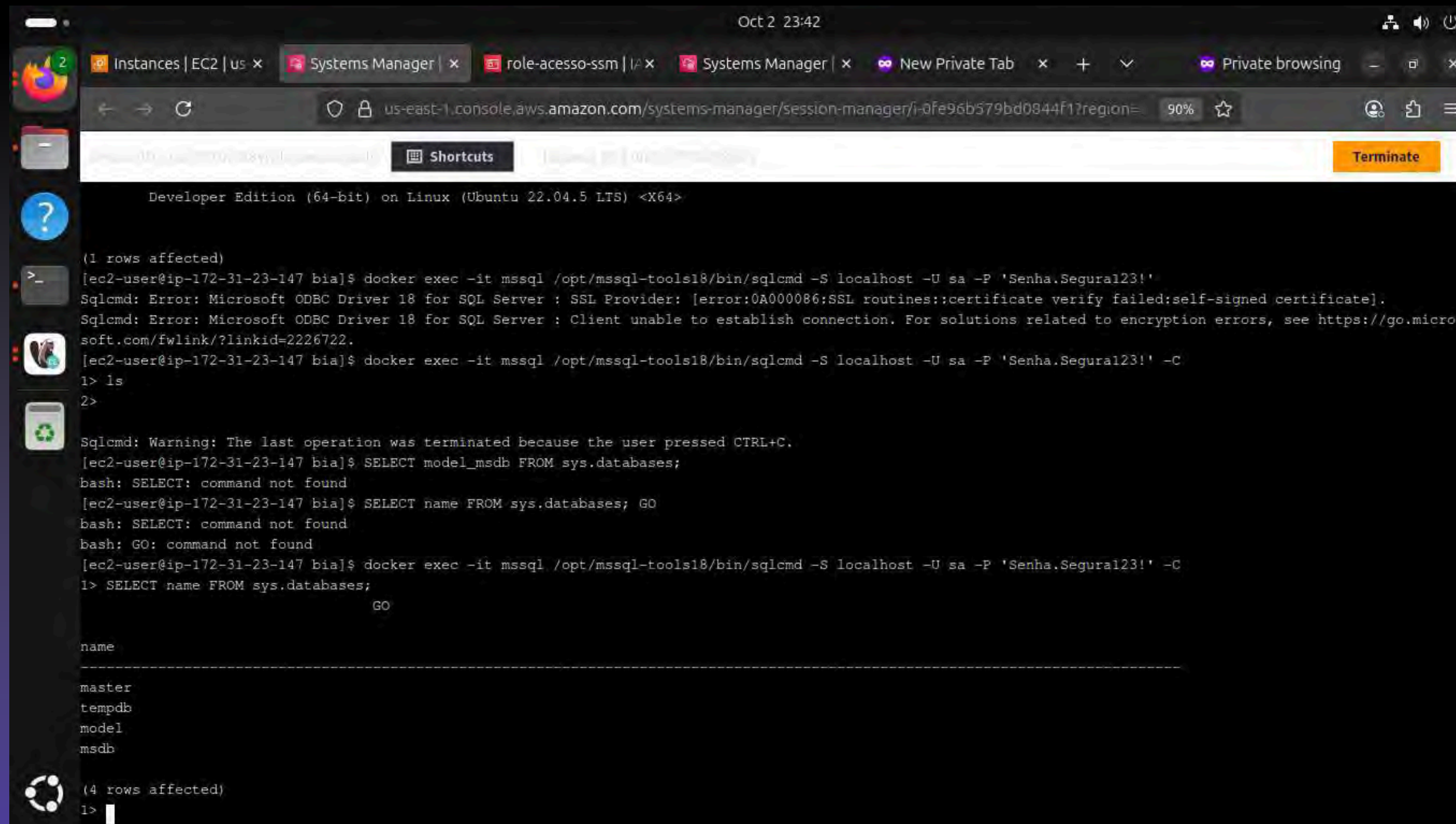
```
Sep 28 00:52
ec2-user@ip-172-31-23-147:~
[ec2-user@ip-172-31-23-147 ~]$ docker rm -f sql1 2>/dev/null || true

docker run -e "ACCEPT_EULA=Y" \
-e "MSSQL_SA_PASSWORD=Enz0l2004@" \
-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 ~]$
```


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.

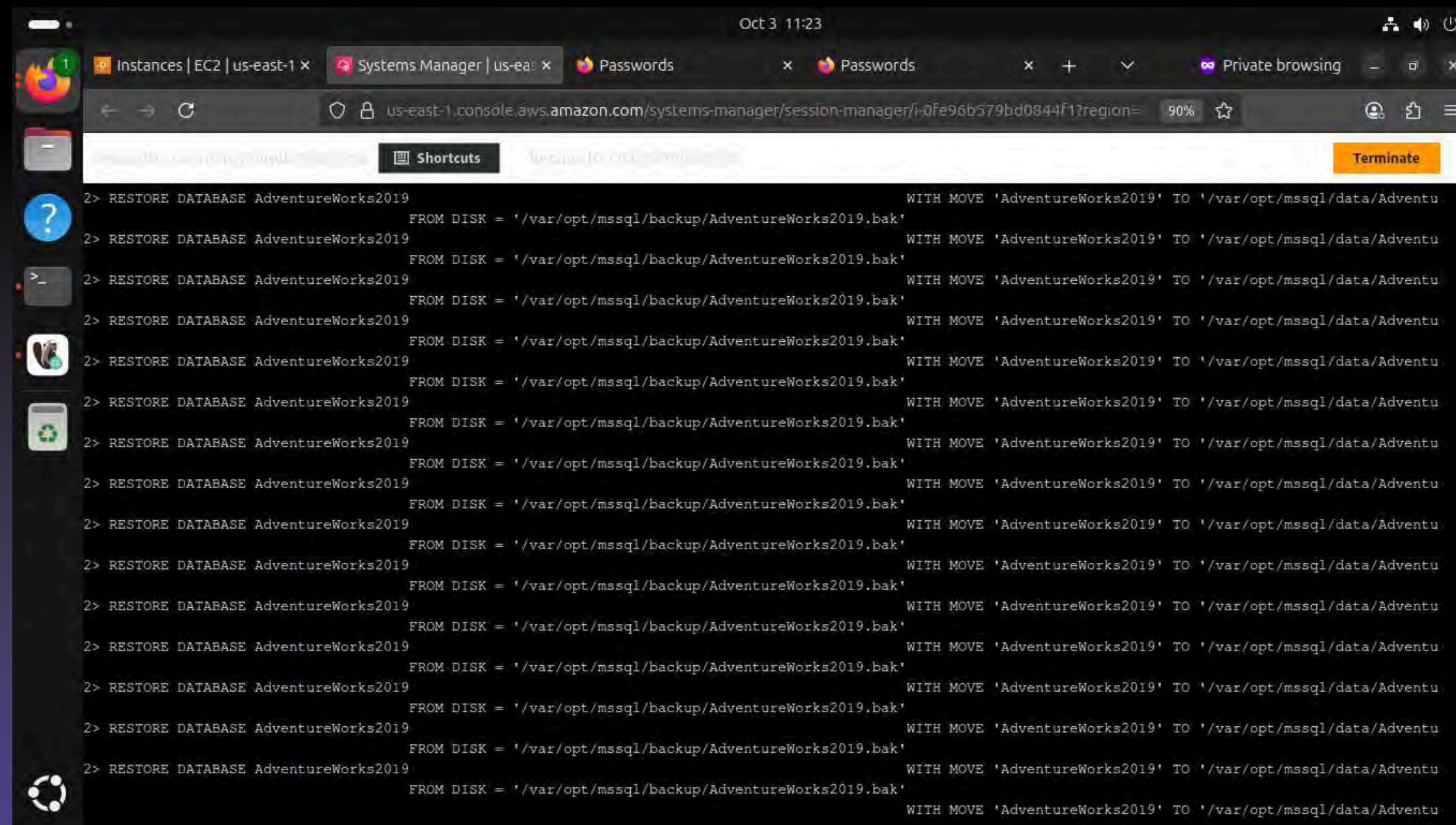


```
Oct 2 23:42
Instances | EC2 | us- x Systems Manager | x role-acesso-ssm | x Systems Manager | x New Private Tab x + v Private browsing
us-east-1.console.aws.amazon.com/systems-manager/session-manager/i-0fe96b579bd0844f1?region= 90% ☆
Shortcuts Terminate
Developer Edition (64-bit) on Linux (Ubuntu 22.04.5 LTS) <X64>
(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.Segural23!'
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.micro
soft.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.Segural23!' -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.Segural23!' -C
1> SELECT name FROM sys.databases;
GO
name
-----
master
tempdb
model
msdb
(4 rows affected)
1>
```

15. Restauração do Banco AdventureWorks2019

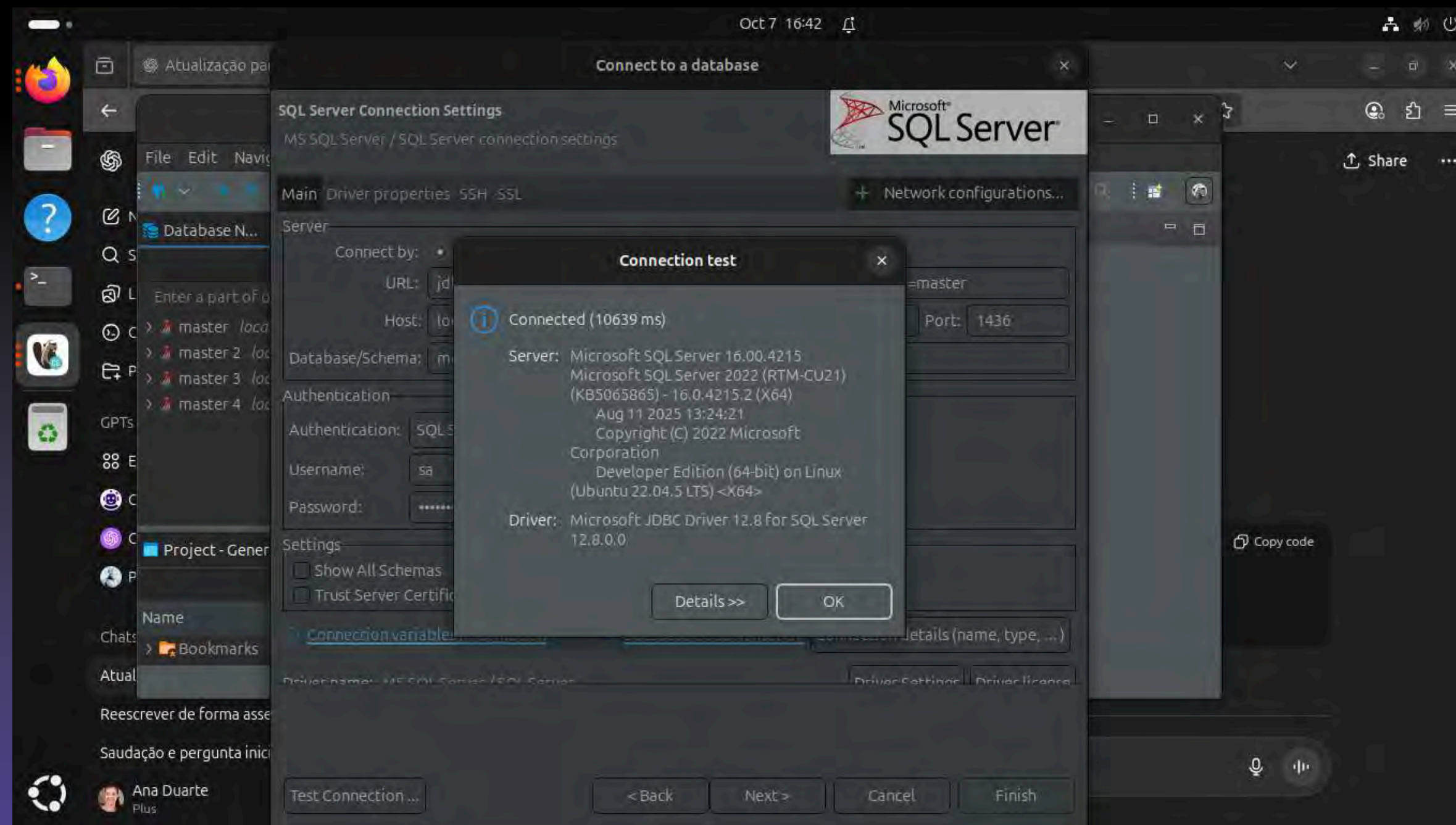
A restauração foi realizada a partir do backup em /var/opt/mssql/backup/.

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

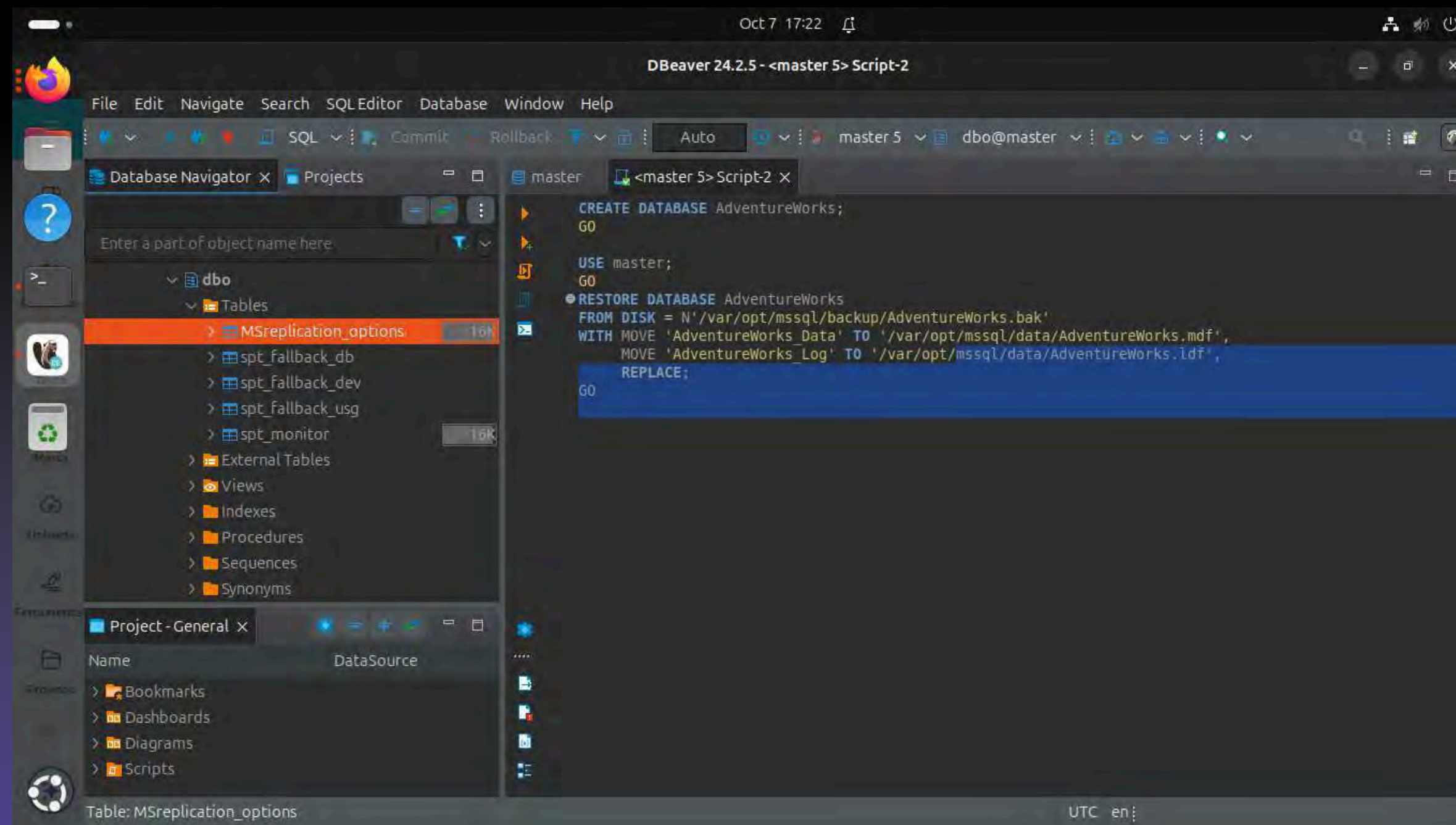


17. A conexão com o SQL Server 2022 foi realizada com sucesso na porta 1436, validando o mapeamento correto entre o host (Ubuntu/VM) e o container Docker.

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.



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

