



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Projeto de Infraestrutura

Colaboradores: Davih G. Duque, Fabiana S. Soares, Filipe A. Costa,
Leonardo G. G. Junior, Vítor C. R. Francisco, Lucas O. Fonseca

Orientador: Fabio L. R. Cordeiro

Resumo

O projeto apresenta o desenvolvimento de um plano completo de infraestrutura de rede para a Universidade Polaris, composta por quatro unidades interligadas e complementares localizadas em Belo Horizonte, Betim, Contagem e Nova Lima. A proposta busca atender à necessidade de integração entre polos com diferentes finalidades acadêmicas e tecnológicas, garantindo conectividade, segurança e desempenho nos serviços prestados. A justificativa fundamenta-se na importância de planejar e implantar uma arquitetura de rede moderna, escalável e segura, capaz de sustentar as operações educacionais e administrativas da instituição. O objetivo geral consiste em projetar e implementar uma solução de infraestrutura híbrida, combinando servidores locais e em nuvem, atendendo às demandas iniciais da Universidade Polaris.

Palavras-chave: infraestrutura de rede; universidade; servidores; nuvem; segurança da informação.

1 INTRODUÇÃO

Este documento apresenta os serviços de infraestrutura de rede implementados para a Universidade Polaris, contemplando tanto recursos locais (on-premise) quanto soluções em nuvem. O objetivo é registrar e detalhar a configuração dos serviços, a estrutura de autenticação centralizada, o gerenciamento de usuários e grupos, bem como a aplicação de políticas de acesso, permitindo uma visão completa da arquitetura implantada. Este relatório serve como referência técnica para acompanhamento, manutenção e futuras expansões da rede, garantindo que todos os serviços estejam organizados, seguros e alinhados às necessidades acadêmicas e administrativas da instituição.

2 DESENVOLVIMENTO

2.1 Serviços on-premise

2.1.1 DHCP

O serviço *Dynamic Host Configuration Protocol* (DHCP) foi configurado com o objetivo de automatizar a atribuição de endereços IP e demais parâmetros de rede aos dispositivos conectados na infraestrutura virtual. Para este experimento, o servidor DHCP foi implementado em uma máquina virtual Linux (Ubuntu Server), hospedada no VirtualBox, e destinada exclusivamente à distribuição de endereços na rede interna do ambiente.

O cenário foi composto por duas máquinas virtuais: uma atuando como servidor DHCP e outra como cliente. O servidor foi configurado com dois adaptadores de rede: o primeiro, do tipo *NAT*, responsável pelo acesso à internet, e o segundo, configurado como *Rede Interna*, designado para prover serviços aos clientes locais. Nesta interface interna foi atribuído o endereço IP estático 192.168.90.1/24, servindo como gateway para os dispositivos da rede.

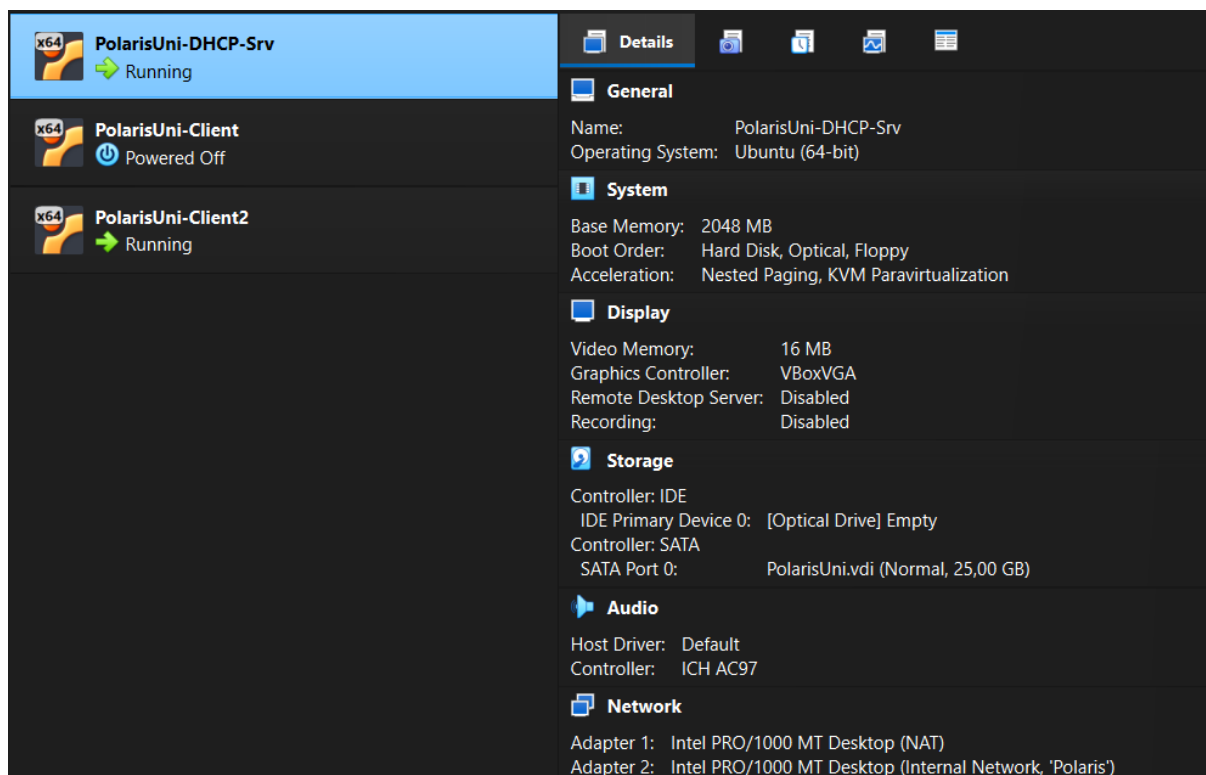


Figura 1 – Janela dos adaptadores de rede configurados.

A instalação do serviço foi realizada por meio do pacote `isc-dhcp-server`, amplamente utilizado em sistemas baseados em Debian. Após a instalação, o arquivo de configuração `/etc/default/isc-dhcp-server` foi ajustado para definir a interface de atuação do serviço, conforme o exemplo a seguir:

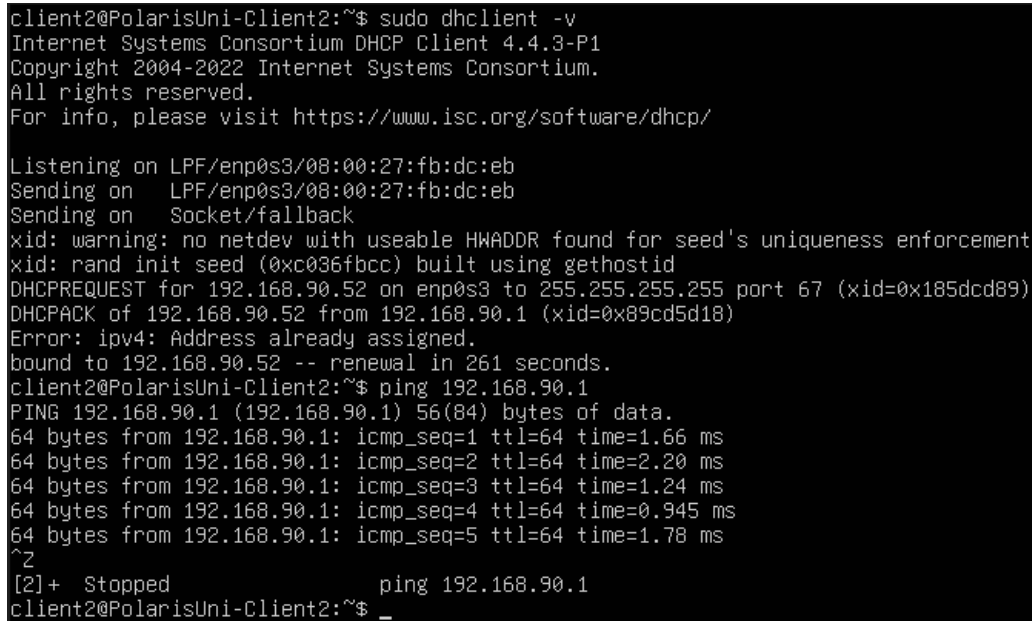
```
INTERFACESv4="enp0s8"
```

Em seguida, o escopo de distribuição de endereços foi especificado no arquivo principal de configuração, localizado em `/etc/dhcp/dhcpd.conf`, com os seguintes parâmetros:

```
subnet 192.168.90.0 netmask 255.255.255.0 {
    range 192.168.90.50 192.168.90.100;
    option routers 192.168.90.1;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 8.8.8.8, 1.1.1.1;
    default-lease-time 600;
    max-lease-time 7200;
}
```

Essa configuração define a faixa de endereços IP a ser distribuída aos clientes, o roteador padrão (gateway) e os servidores DNS utilizados para resolução de nomes. Após a configuração, o serviço foi reiniciado e verificado com o comando `systemctl status isc-dhcp-server`, confirmando seu estado ativo e em execução.

No cliente, também executando em uma máquina virtual conectada à mesma rede interna, o teste de funcionamento foi realizado com o comando `sudo dhclient -v`. O dispositivo obteve automaticamente um endereço IP dentro do intervalo configurado (por exemplo, 192.168.90.51), comprovando o correto funcionamento do servidor DHCP. A comunicação entre as máquinas foi validada com sucesso através do comando `ping 192.168.90.1`, evidenciando que o servidor estava respondendo e que a rede estava operacional.

A terminal window showing the execution of the 'dhclient -v' command. The output displays the DHCP client's initialization, including listening on the network interface, sending requests, and receiving a response from the server at 192.168.90.1. It also shows the assignment of the IP address 192.168.90.52. Following this, the 'ping 192.168.90.1' command is executed, showing five successful ping requests with response times ranging from approximately 0.945 ms to 2.20 ms.

```
client2@PolarisUni-Client2:~$ sudo dhclient -v
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp0s3/08:00:27:fb:dc:eb
Sending on   LPF/enp0s3/08:00:27:fb:dc:eb
Sending on   Socket/fallback
xid: warning: no netdev with useable HWADDR found for seed's uniqueness enforcement
xid: rand init seed (0xc036fbcc) built using gethostid
DHCPREQUEST for 192.168.90.52 on enp0s3 to 255.255.255.255 port 67 (xid=0x185dcd89)
DHCPACK of 192.168.90.52 from 192.168.90.1 (xid=0x89cd5d18)
Error: ipv4: Address already assigned.
bound to 192.168.90.52 -- renewal in 261 seconds.
client2@PolarisUni-Client2:~$ ping 192.168.90.1
PING 192.168.90.1 (192.168.90.1) 56(84) bytes of data.
64 bytes from 192.168.90.1: icmp_seq=1 ttl=64 time=1.66 ms
64 bytes from 192.168.90.1: icmp_seq=2 ttl=64 time=2.20 ms
64 bytes from 192.168.90.1: icmp_seq=3 ttl=64 time=1.24 ms
64 bytes from 192.168.90.1: icmp_seq=4 ttl=64 time=0.945 ms
64 bytes from 192.168.90.1: icmp_seq=5 ttl=64 time=1.78 ms
^Z
[2]+  Stopped                  ping 192.168.90.1
client2@PolarisUni-Client2:~$ _
```

Figura 2 – Janela mostrando o ping do client funcionando.

A implementação do DHCP nesse ambiente virtual demonstra a eficiência da configuração automática de parâmetros de rede em sistemas isolados, permitindo maior escalabilidade e controle sobre os endereços IP distribuídos. Além disso, o uso de um servidor dedicado para a função promove uma topologia mais próxima da adotada em ambientes corporativos reais, com separação clara entre as funções de roteamento, fornecimento de serviços e comunicação com a internet.

2.1.2 Active Directory

O *Active Directory* (AD) é um serviço de diretório desenvolvido pela Microsoft que tem como principal função gerenciar identidades, autenticações e permissões dentro de um ambiente de rede. Ele permite o controle centralizado de usuários, grupos, computadores e políticas de segurança, facilitando a administração de recursos em redes corporativas e institucionais. Através de sua estrutura hierárquica e organizada em domínios, o Active Directory garante maior eficiência e segurança na autenticação dos usuários, além de possibilitar o gerenciamento de acessos de forma padronizada.

No contexto da Universidade Polaris, a implementação do Active Directory foi fun-

damental para estabelecer um sistema de autenticação unificado entre as unidades. Com ele, é possível que alunos, professores e colaboradores acessem os recursos da rede utilizando credenciais únicas, garantindo controle e rastreabilidade das atividades. Além disso, o AD contribui para a padronização das políticas de segurança e a redução de falhas decorrentes de configurações isoladas em cada campus. Assim, o serviço atua como uma base sólida para a infraestrutura de rede proposta, promovendo integração e segurança em todo o ambiente institucional.

Para configurar o ambiente prático, foi utilizada uma máquina virtual no *Windows Server*, realizando o mapeamento dos campi e a criação das Unidades Organizacionais (OUs) de cada setor, organizadas em **Administrativo**, **Corpo Docente** e **Colegiado**. Cada setor recebeu políticas de acesso específicas, definidas da seguinte forma:

- **Administrativo:** topo da pirâmide, com acesso irrestrito e capacidade de delegação e ajustes globais na rede;
- **Corpo Docente:** acesso intermediário para atualizar sistemas e utilizar recursos gerais de ensino;
- **Colegiado:** acesso restrito apenas ao essencial, garantindo que os alunos permaneçam focados nas atividades acadêmicas.

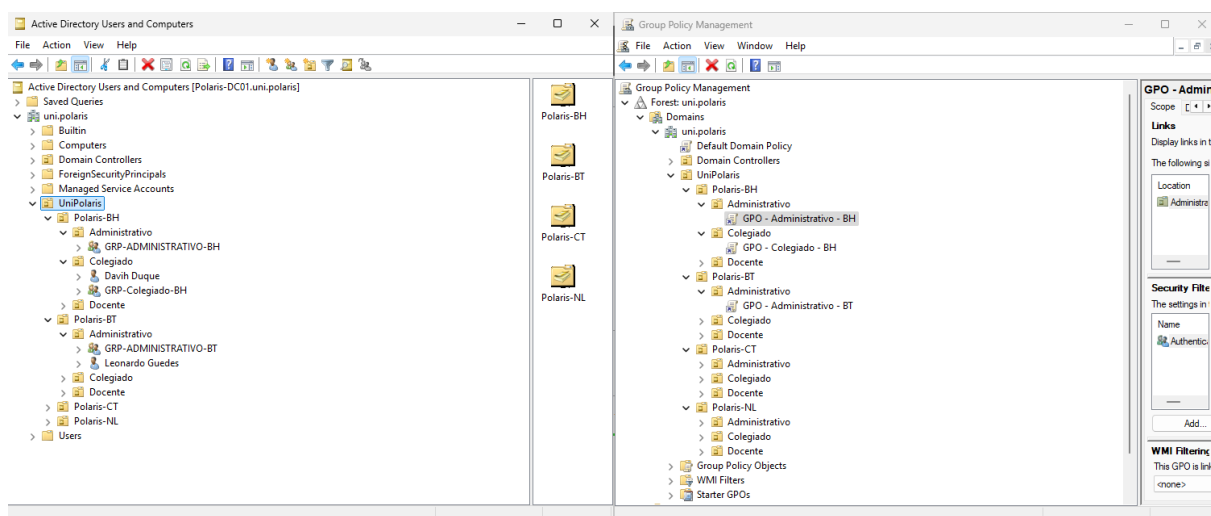


Figura 3 – Janela do Active Directory Users and Computers com as OUs e políticas configuradas.

2.1.3 DNS

O *Domain Name System* (DNS) é um serviço de rede responsável pela tradução de nomes de domínio em endereços IP, permitindo que computadores, servidores e outros dispositivos localizados na rede se comuniquem de forma eficiente. No contexto do Active Directory, o DNS é fundamental, pois os controladores de domínio dependem dele para localizar e autenticar usuários, computadores e serviços dentro do domínio.

Na Universidade Polaris, o DNS foi configurado no mesmo servidor onde o Active Directory foi implementado, garantindo integração completa com o domínio da instituição. Durante a promoção do servidor a Controlador de Domínio, o serviço de DNS foi instalado e configurado para gerenciar a zona principal do domínio, permitindo que todas as unidades da universidade resolvam nomes de host e serviços de forma centralizada. Esta configuração assegura que os registros de computadores, controladores de domínio e grupos estejam sempre atualizados e disponíveis, facilitando a comunicação interna e mantendo a rede organizada e segura.

Na prática, foram configuradas zonas primárias e registros essenciais, garantindo que:

- Os controladores de domínio possam ser localizados corretamente por toda a rede;
- Computadores e usuários possam autenticar e acessar serviços sem problemas;
- A manutenção e a atualização de registros de rede sejam centralizadas e padronizadas.

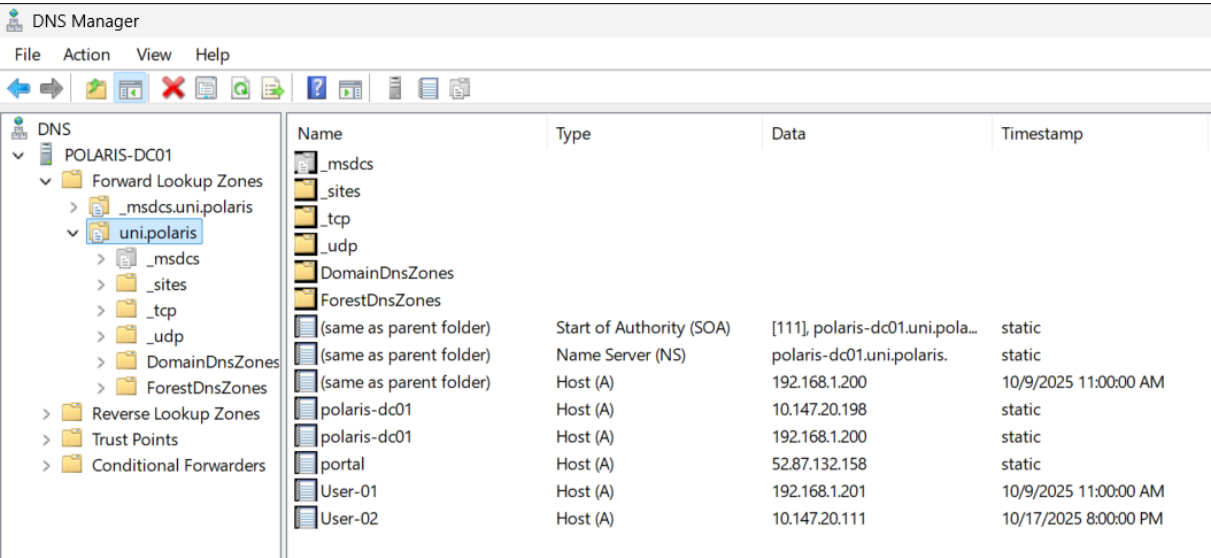


Figura 4 – Configuração do DNS no Windows Server, exibindo a zona principal do domínio e os registros vinculados ao Active Directory.

2.2 Serviços em Nuvem

2.2.1 HTTP

O serviço *HTTP* foi configurado com o objetivo de permitir hospedagem das aplicações Web da instituição. A implementação foi realizada em uma máquina virtual *Ubuntu Server 24.04*, hospedada em uma instância do *EC2*, dentro do ambiente *Amazon AWS*, representando o servidor HTTP principal da infraestrutura.

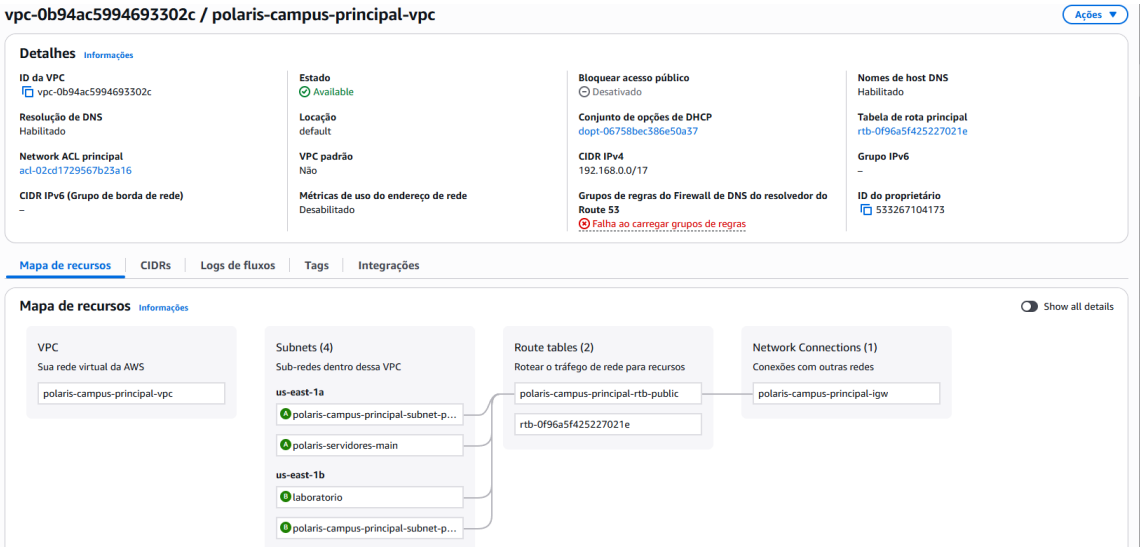


Figura 5 – Configurações da VPC para servidor WEB

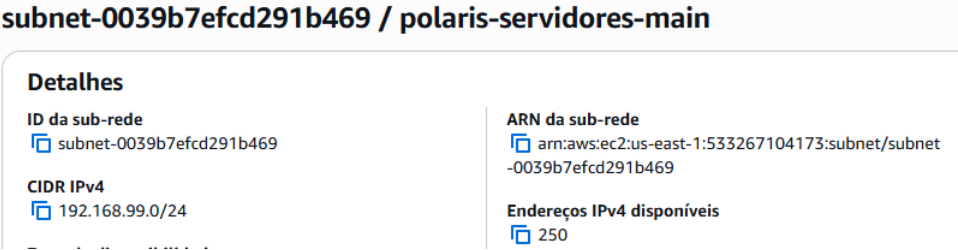


Figura 6 – Configurações da Sub-rede para servidor WEB

Foi configurada uma VPC e uma sub-rede emulando as do campus principal da Polaris, respectivamente 192.168.0.0/17 para o campus Belo Horizonte e 192.168.99.0/24 pra os servidores.

Um gateway foi associado À VPC, permitindo acesso dessa sub-rede à internet.A *Route Table* (tabela de rotas da sub-rede pública foi configurada para direcionar todo o tráfego externo (0.0.0.0/0) para o *Internet Gateway*, garantindo o acesso via IP público.



Figura 7 – Configurações da Tabela de rotas para servidor WEB

A segurança de acesso foi controlada por um *Security Group*, que foi configurado para permitir tráfego de entrada e saída público nas portas 80, por protocolo HTTP, e 22, por protocolo SSH.

A instância contendo a máquina Ubuntu foi configurada com um IPv4 Elástico associado

rtb-0e638338eee534ae5 / polaris-campus-principal-rtb-public Ações

Detalhes Informações ID da tabela de rotas rtb-0e638338eee534ae5 VPC vpc-0b94ac5994693302c polaris-campus-principal-vpc	Principal Não ID do proprietário 533267104173	Associações explícitas de sub-rede 4 sub-redes	Associações de borda -
---	--	---	---------------------------

[Rotas](#) | [Associações de sub-rede](#) | [Associações de borda](#) | [Propagação de rotas](#) | [Tags](#)

Rotas (2) Ambos Editar rotas

Destino	Alvo	Status	Propagado	Route Origin
0.0.0.0/0	igw-0b0b2bce5d986d0b6	Ativo	Não	Create Route
192.168.0.0/17	local	Ativo	Não	Create Route Table

Figura 8 – Configurações da Tabela de rotas para servidor WEB

52.87.132.158 é um DNS públicos fornecidos pelo provedor de de nuvem e com um ip privado de 192.168.99.4, conforme documentação. O servidor Web foi hospedado utilizando-se o *Apache*, expondo o port 80 - por padrão o responsável pela comunicação HTTP - para comunicação HTTP pelo ip público da máquina.

Resumo da instância para i-0d4a495e9ce6655af (polaris-servidor-http) Informações Conectar Estado da instância Ações

Atualizado há less than a minute ID da instância i-0d4a495e9ce6655af Endereço IPv6 - Tipo de nome do host Nome do IP: ip-192-168-99-4.ec2.internal Nome do DNS do recurso privado de resposta - Endereço IP atribuído automaticamente - Função do IAM -	Endereço IPv4 público 52.87.132.158 endereço aberto Estado da instância Executando Nome do DNS de IP privado (somente IPv4) ip-192-168-99-4.ec2.internal Tipo de instância t3.micro ID da VPC vpc-0b94ac5994693302c (polaris-campus-principal-vpc) ID da sub-rede subnet-0039b7efcd291b469 (polaris-servidores-main)	Endereços IPv4 privados 192.168.99.4 DNS pública ec2-52-87-132-158.compute-1.amazonaws.com endereço aberto Endereços IP elásticos 52.87.132.158 [IP público] Descoberto do AWS Compute Optimizer Opte por participar do AWS Compute Optimizer para obter recomendação s. Saiba mais Nome do Grupo do Auto Scaling -
---	---	---

Figura 9 – Configurações da instância do servidor WEB

Em seguida, usou-se SCP para enviar os arquivos de implementação da aplicação principal da Universidade, uma landing page simples que leva à documentação do projeto de redes. Esse app foi adicionado ao index dentro da pasta principal do Apache2 /var/www/html.

Pode adicionar-se mais apps e arquivos ao sistema Web da Universidade adicionando arquivos a esse mesmo diretório dentro do servidor conectando-se a ele por SSH ou acessando diretamente a máquina do servidor.

Credenciais de acesso do servidor HTTP:

- Usuário: ubuntu
- Senha: polaris#1234

O código para essa aplicação disponível [neste repositório](#).

A landing page mostrou-se acessível tanto pelo DNS da universidade quando pelo seu ip público, ambos usando protocolo HTTP.

Em um cenário real de melhoria contínua, poderia considerar-se:

- Implementação do protocolo HTTPS para serviços públicos



Figura 10 – Aplicação WEB: Landing Page polaris

- Implementar uma lógica de Integração e desenvolvimento contínuo no repositório do Github, permitindo testes unitários e de ponta a ponta e deploy da aplicação de forma automatizada, em uma pipeline integrada à instância.

2.2.2 Base de dados

Para suportar as aplicações da instituição com um serviço de banco de dados robusto, foi implementado um Sistema Gerenciador de Banco de Dados (SGBD) utilizando o serviço *Amazon RDS (Relational Database Service)*.

Escolheu-se o sistema de gerenciador **MariaDB**. Esta decisão foi justificada por ser um projeto *open-source*, derivado (fork) do reconhecido *MySQL*. Portanto, o MariaDB possui confiável escalabilidade e compatibilidade com o ecossistema *MySQL*.

A instância do banco de dados foi provisionada na mesma sub-rede dos servidores principais (192.168.99.0/24), dentro da VPC do campus Belo Horizonte. Para controlar o acesso, foi criado um *Security Group* específico para o banco de dados, com uma regra que acesso na porta 3306 (padrão do protocolo *MySQL/MariaDB*) a partir de toda WEB e outra que permite ssh na rede interna da universidade.

Ao final da configuração, a AWS provisionou um *endpoint* que serve como o endereço exclusivo para conexões com o banco de dados pelo protocolo *MySQL/MariaDB*.

Para validar o funcionamento, foi realizado um teste de conexão a partir da instância EC2 que hospeda o servidor web. Primeiramente, instalou-se um cliente MariaDB na máquina Ubuntu:

```
sudo apt install mariadb-client
```

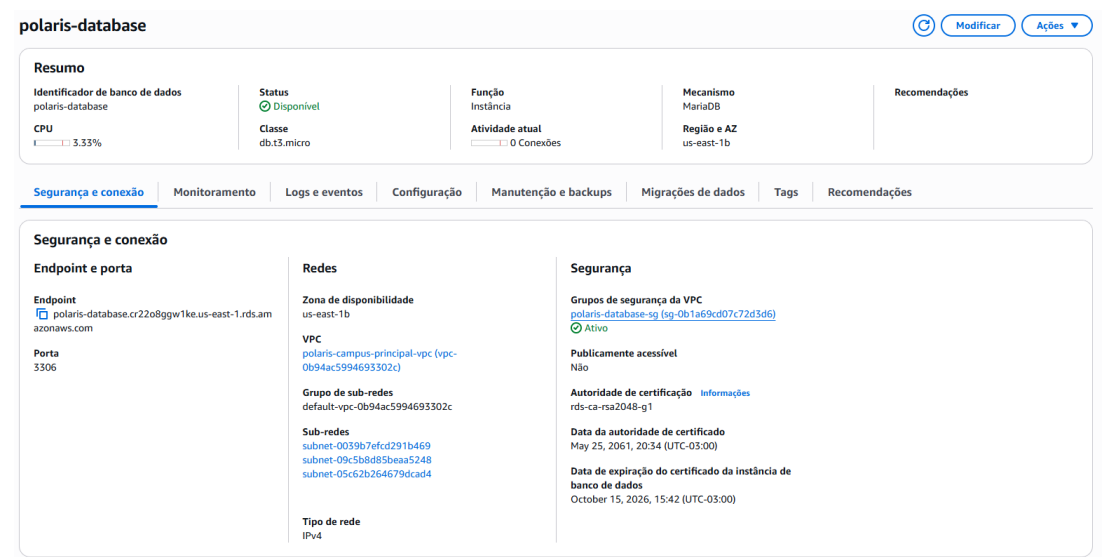


Figura 11 – Configurações da Instância do banco de dados

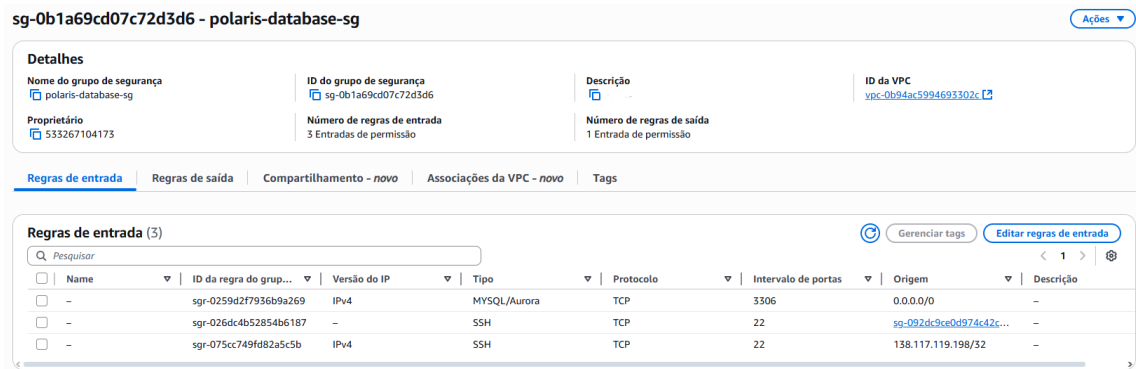


Figura 12 – Configurações da Instância do banco de dados

Em seguida, foi executado o comando de conexão utilizando o endpoint, o usuário e a senha definidos durante a criação da instância RDS:

```
mysql -h polaris-db.xxxxxxxxx.us-east-1.rds.amazonaws.com -u admin -p
```

Credenciais da Base de Dados:

- Usuário: polaris_admin
- Senha: polaris_admin#1234

Após a conexão bem-sucedida, selecionou-se a base de dados para uso a partir do CLI do MariaDb, confirmando que a infraestrutura estava operacional.

Em um cenário real, diversas melhorias seriam cruciais, baseadas em consensos acadêmicos e arquiteturas de referência sobre segurança e alta disponibilidade:

- **Alta Disponibilidade (Multi-AZ):** Para sistemas críticos como portais acadêmicos ou sistemas de matrícula, a implantação em modo *Multi-AZ* é fundamental. O RDS automaticamente provisiona e mantém uma réplica síncrona do banco de dados em uma Zona de Disponibilidade diferente, garantindo um *failover* automático em caso de falha da instância primária.

- **Gerenciamento de Credenciais:** Em vez de expor senhas em scripts ou arquivos de configuração, as credenciais do banco de dados deveriam ser armazenadas e rotacionadas de forma segura utilizando o *AWS Secrets Manager*.
- **Escalabilidade com Réplicas de Leitura:** Para aplicações com alta demanda de leitura (ex: consulta de notas, catálogos de cursos), a criação de *Read Replicas* permitiria desviar esse tráfego da instância principal, melhorando a performance geral do sistema.

2.2.3 FTP com armazenamento EFS na AWS

O serviço *File Transfer Protocol* (FTP) foi configurado com o objetivo de permitir o compartilhamento de arquivos de forma segura e persistente entre os servidores da Universidade Polaris. Para garantir alta disponibilidade e escalabilidade, o serviço foi integrado ao *Amazon Elastic File System* (EFS), possibilitando que múltiplas instâncias EC2 acessem o mesmo diretório de armazenamento na nuvem.

A implementação foi realizada em uma instância *Ubuntu Server 22.04* hospedada na AWS EC2, com as seguintes características principais:

- Tipo da instância: `t3.small`;
- IP público: `13.221.25.115`;
- Sistema de arquivos EFS montado em `/mnt/efs`;
- Usuário local de acesso: `aluno` (senha: `123`);
- Protocolo de transferência: FTP (porta 21).

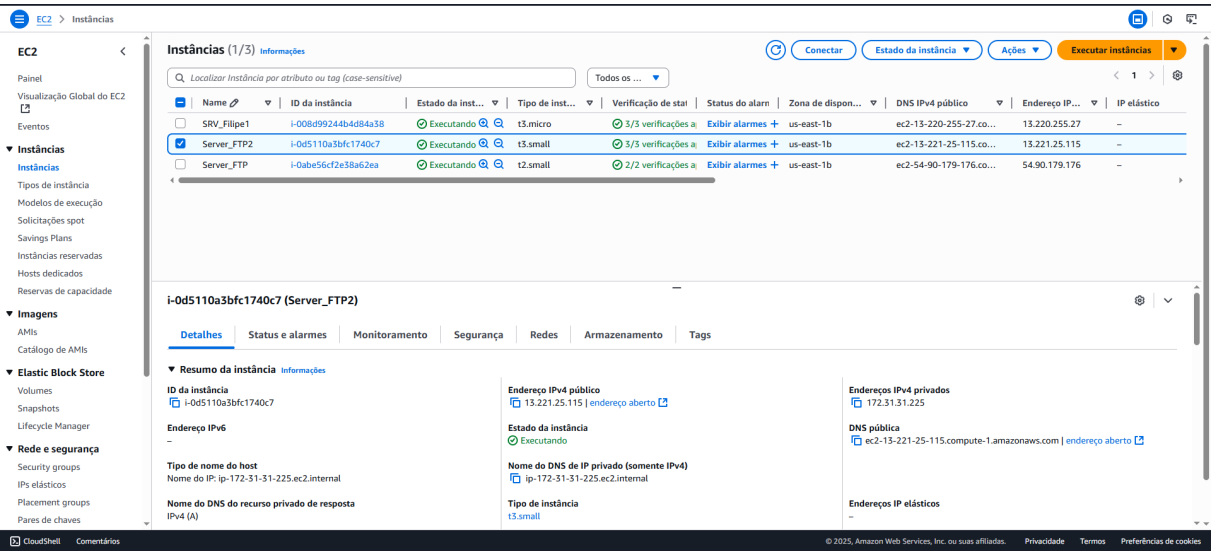


Figura 13 – Instância FTP

2.2.3.1 Criação e configuração da instância EC2

A instância EC2 foi criada e conectada via SSH através do comando:

```
ssh -i ftpserverkey2.pem ubuntu@13.221.25.115
```

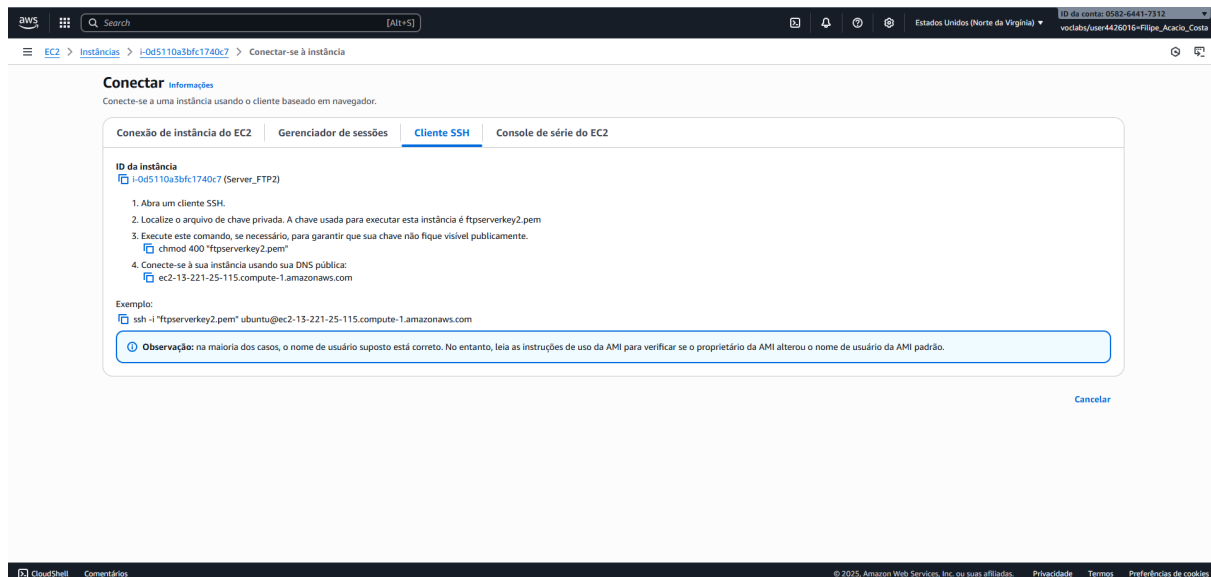


Figura 14 – Conexão Instância SSH

2.2.3.2 Instalação e configuração do servidor FTP

O serviço *vsFTPD* foi instalado e configurado no servidor com os comandos:

```
sudo apt update && sudo apt upgrade -y
sudo apt install vsftpd -y
```

Em seguida, o arquivo de configuração principal `/etc/vsftpd.conf` foi ajustado para definir o diretório raiz dos usuários e aplicar as políticas de isolamento, conforme o trecho a seguir:

```
local_root=/mnt/efs/aluno
chroot_local_user=YES
allow_writeable_chroot=YES
```

O diretório de trabalho do usuário `aluno` foi configurado com as permissões adequadas:

```
sudo chown root:root /mnt/efs/aluno
sudo chmod 755 /mnt/efs/aluno
sudo mkdir /mnt/efs/aluno/files
sudo chown aluno:aluno /mnt/efs/aluno/files
```

Após as alterações, o serviço foi reiniciado:

```
sudo systemctl restart vsftpd
```

2.2.3.3 Criação e montagem do EFS

O sistema de arquivos *Amazon EFS* foi criado com o nome `FTP`, utilizando o modo de desempenho *Uso Geral*, criptografia ativada e backups automáticos. O EFS foi associado à mesma VPC da instância EC2, com regras de segurança permitindo tráfego NFS (porta 2049) entre os dois recursos.

A montagem do EFS na instância foi feita através dos comandos:

```
sudo mkdir -p /mnt/efs
sudo mount -t nfs4 -o nfsvers=4.1 fs-02896c5c15e8de3b7.efs.us-east-1.am
```

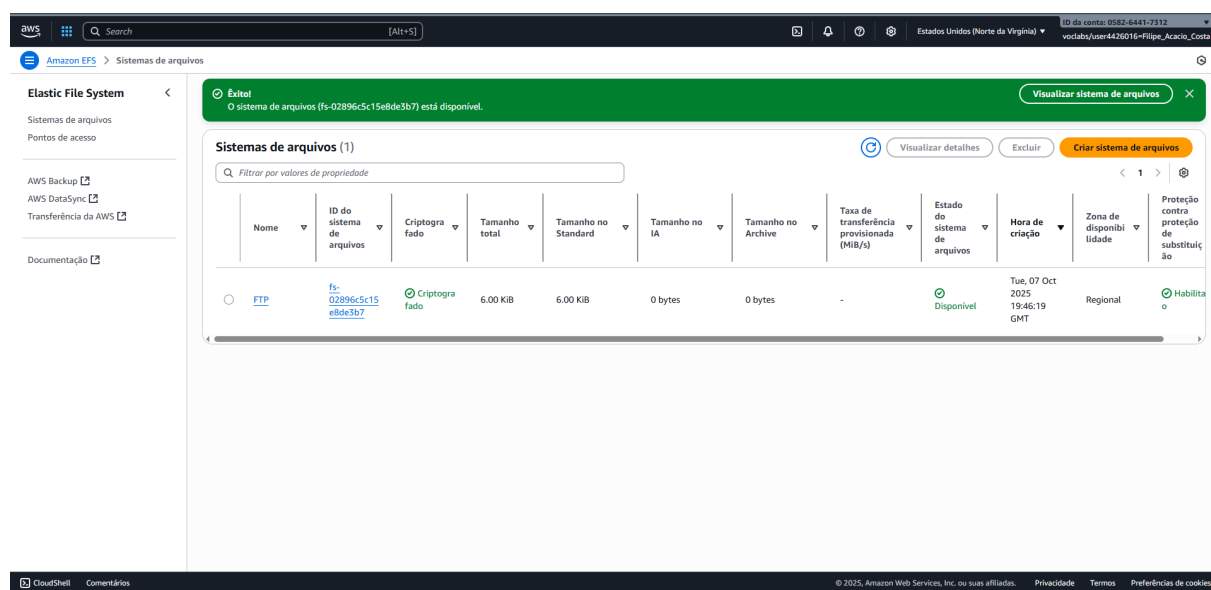


Figura 15 – Sistema de Arquivos Amazon EFS

A verificação foi realizada utilizando o comando:

```
df -h
```

O retorno confirmou a presença do EFS como sistema de arquivos montado em `/mnt/efs`. Para garantir a montagem automática na inicialização, a entrada a seguir foi adicionada ao arquivo `/etc/fstab`:

```
fs-02896c5c15e8de3b7.efs.us-east-1.amazonaws.com:/ /mnt/efs
nfs4 defaults,_netdev 0 0
```

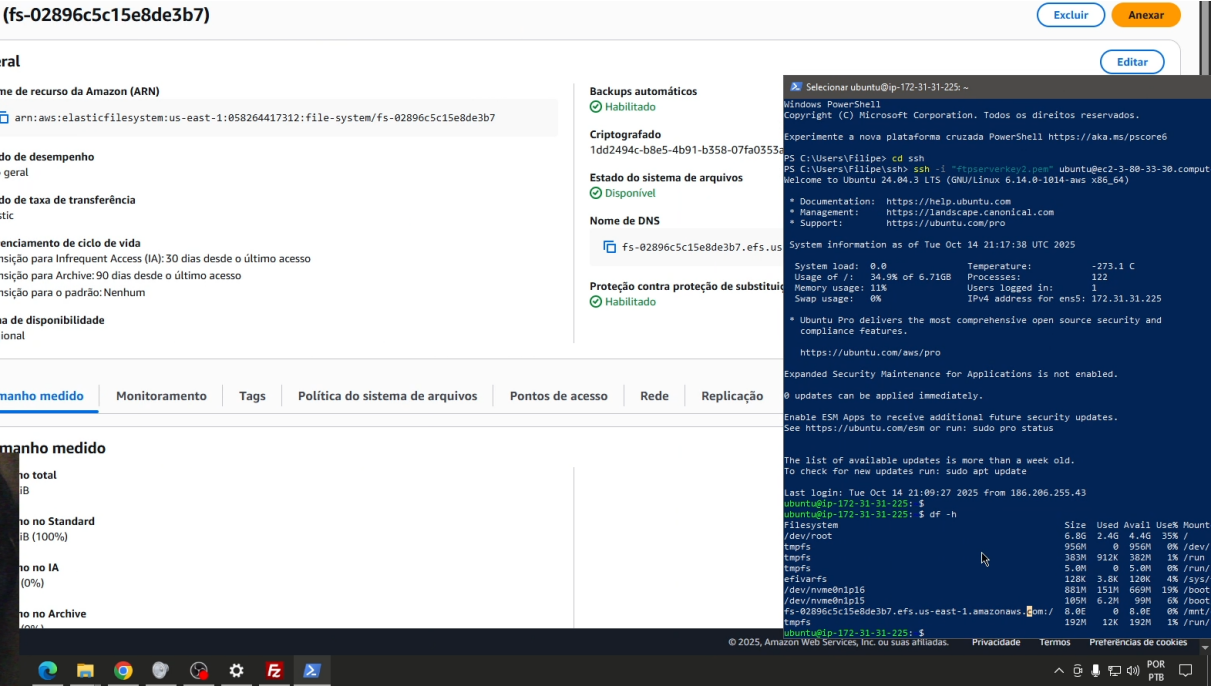


Figura 16 – Verificação da montagem do Sistema de Arquivos Amazon EFS

2.2.3.4 Ajustes de segurança e conectividade

O *Security Group* associado à instância EC2 (grupo default) foi configurado para permitir as seguintes regras de entrada:

- TCP 20–21: conexões FTP padrão;
- TCP 2049: comunicação entre a instância EC2 e o sistema de arquivos Amazon EFS;
- TCP 990: conexões FTPS (caso habilitado);
- TCP 30000–31000: portas do modo passivo FTP;
- TCP 22: acesso SSH para administração.

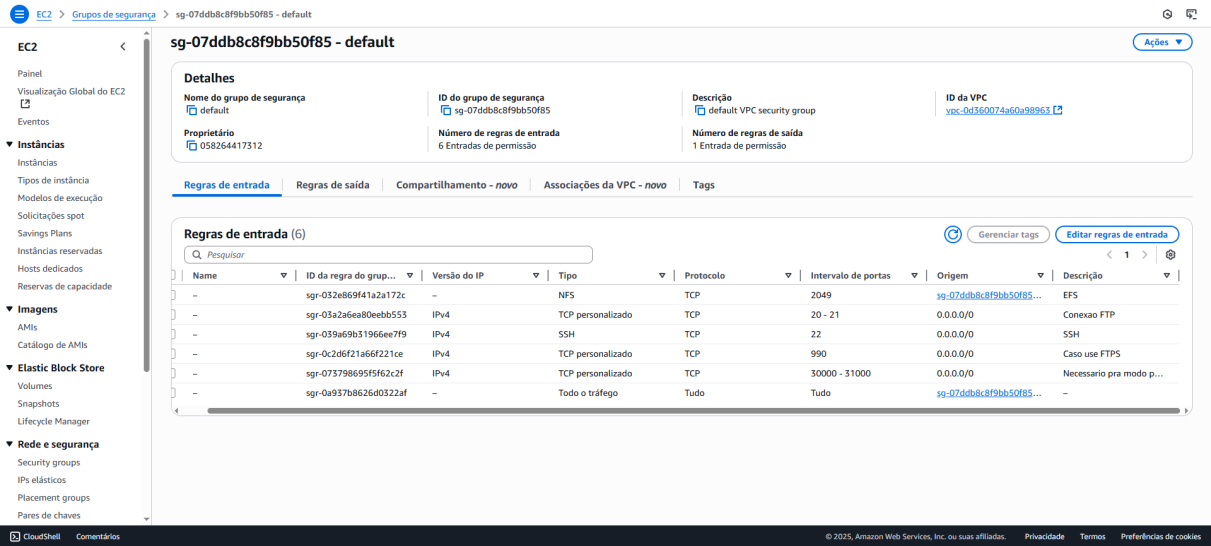


Figura 17 – Regras de Entrada dos Grupos de Segurança

Essas regras garantem que o cliente (FileZilla ou outro) consiga estabelecer sessões FTP com a instância de forma segura e estável.

2.2.3.5 Testes de operação

Os testes foram realizados com o cliente *FileZilla*, utilizando os seguintes parâmetros:

Host : 13.221.25.115
Usuário: aluno
Senha: 123
Porta: 21

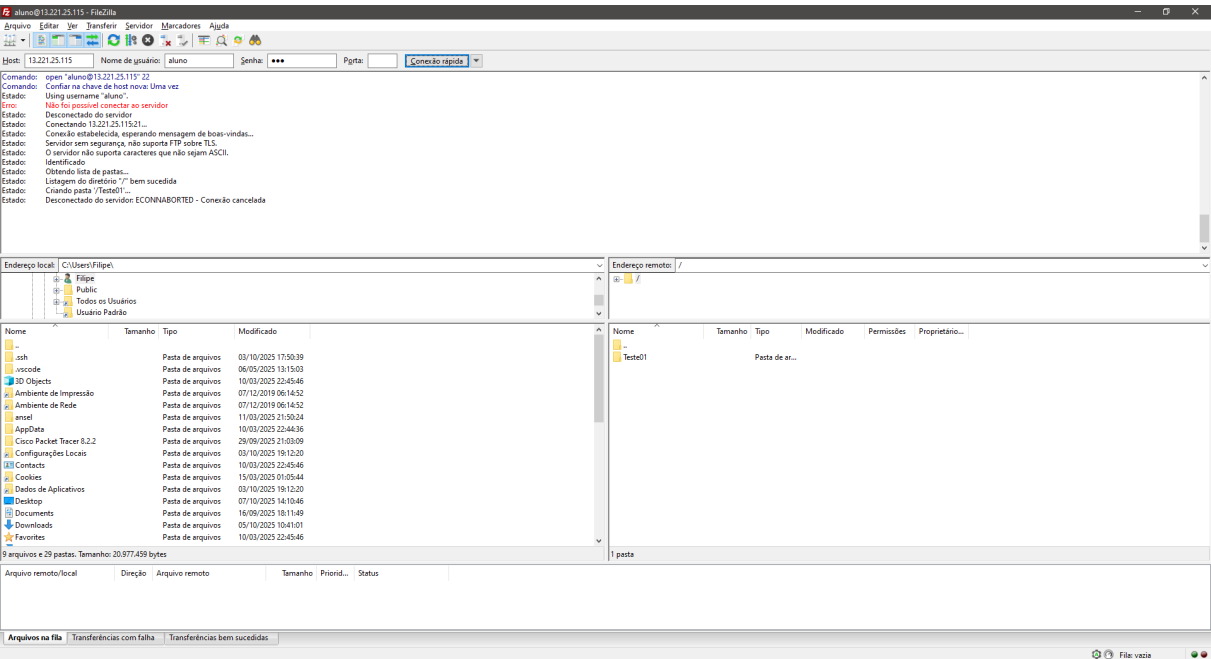


Figura 18 – Teste de operação FTP com Cliente FileZilla

A conexão foi estabelecida com sucesso. No cliente, foi criado o diretório `Teste01`, que apareceu imediatamente na instância, dentro de `/mnt/efs/aluno`, confirmando que o EFS estava sendo utilizado como backend de armazenamento.

```
PS C:\Users\Filipe\ssh>
PS C:\Users\Filipe\ssh> ssh -i "ftpserverkey2.pem" ubuntu@ec2-13-221-25-115.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1014-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Oct  7 21:04:01 UTC 2025

System load:  0.0           Temperature:   -273.1 C
Usage of /:   34.8% of 6.71GB Processes:        128
Memory usage: 13%          Users logged in: 1
Swap usage:   0%           IPv4 address for ens5: 172.31.31.225

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Oct  7 20:16:29 2025 from 186.206.255.43
ubuntu@ip-172-31-31-225:~$
ubuntu@ip-172-31-31-225:~$
ubuntu@ip-172-31-31-225:~$ ls -l /mnt/efs/aluno
total 0
ubuntu@ip-172-31-31-225:~$ ls -l /mnt/efs/aluno
total 4
drwx----- 2 aluno aluno 6144 Oct  7 21:04 Teste01
ubuntu@ip-172-31-31-225:~$ sudo nano /etc/fstab
ubuntu@ip-172-31-31-225:~$ ubuntu@ip-172-31-31-225:~$
ubuntu@ip-172-31-31-225:~$
ubuntu@ip-172-31-31-225:~$ sudo mount -a
ubuntu@ip-172-31-31-225:~$
```

Figura 19 – Listagem no Terminal do diretório FTP criado: Teste01

2.2.3.6 Resultado final e considerações

A integração entre o serviço FTP e o *Amazon EFS* resultou em um ambiente de armazenamento persistente, acessível por múltiplos servidores e com gerenciamento simplificado de usuários. O uso de isolamento via `chroot` garante segurança adicional ao restringir o acesso de cada usuário ao seu próprio diretório.

Essa configuração representa uma solução escalável e segura de armazenamento de arquivos para a infraestrutura acadêmica da Universidade Polaris, integrando serviços em nuvem com protocolos tradicionais de transferência de dados.

3 ANÁLISE DE SEGURANÇA E LOGS - ACTIVE DIRECTORY E DNS

Para garantir a segurança e a auditoria dos serviços críticos da Universidade Polaris, o *Active Directory* (AD) e o DNS foram escolhidos como foco central da configuração de defesa cibernética, devido à grande rotatividade de informações e dados sensíveis que estes serviços gerenciam. Essa abordagem busca reforçar os três pilares da cibersegurança: **disponibilidade**, **confidencialidade** e **integridade**, garantindo proteção mesmo em um ambiente remoto utilizando a rede ZeroTier.

No *Active Directory*, foram habilitados os seguintes tipos de auditoria e logs de segurança:

- **Account Logon** – registros de autenticações de usuários;
- **Account Management** – criação, modificação e exclusão de contas;
- **Logon/Logoff** – controle de sessões de usuários;
- **Privilege Use** – uso de privilégios administrativos;
- **Policy Change** – alterações de políticas de segurança.

A visualização e monitoramento dos eventos é realizada através do *Event Viewer*, permitindo acompanhar atividades suspeitas e auditoria de usuários. Para reforçar a segurança, foi definida uma regra que impede que usuários comuns acessem o servidor localmente, permitindo apenas a um administrador autorizado realizar alterações, garantindo que a criação e gestão de contas não seja comprometida.

O Windows Defender foi configurado com políticas de firewall que restringem o acesso RDP apenas à rede de TI, bloqueando qualquer tentativa de conexão de sub-redes externas, mesmo dentro da mesma rede física. Além disso, foi instalado um servidor *Certification Authority* (CA) para autenticação de dispositivos e habilitado o SMB Signing nas máquinas, reforçando a integridade das comunicações de arquivos.

No serviço de *DNS*, foi habilitado o **DNS Query Logging**, permitindo auditoria detalhada das consultas de nomes de domínio, importante para rastrear possíveis tentativas de acesso indevido.

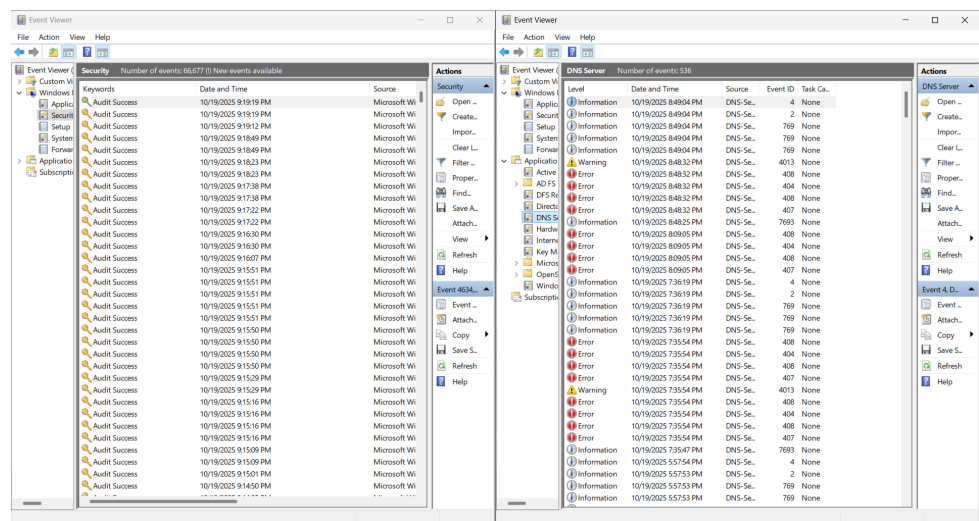


Figura 20 – À esquerda, logs de segurança e auditoria do Active Directory; à direita, logs de consultas do DNS.

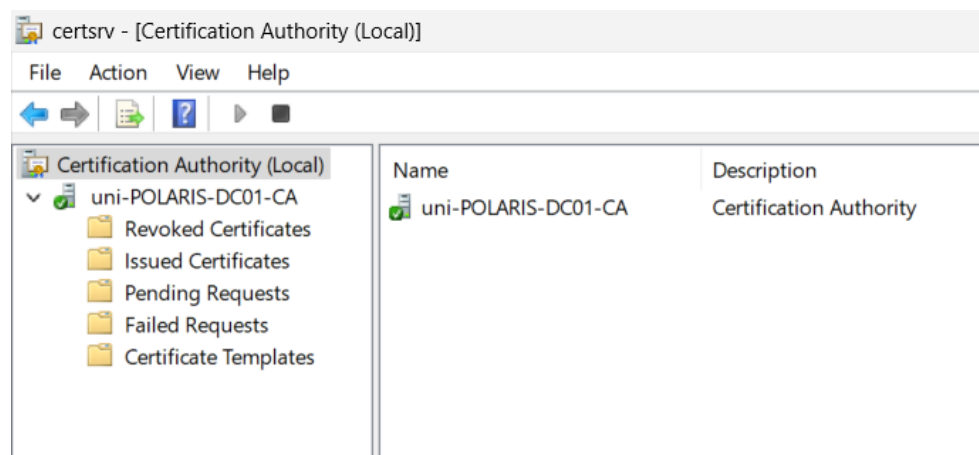


Figura 21 – Certificação de autenticação local do servidor (CA) e configuração do firewall via GPO para restrição de RDP.

4 CONCLUSÃO

A implementação dos serviços de infraestrutura da Universidade Polaris resultou em um ambiente de rede funcional, seguro e integrado entre as unidades. A configuração dos serviços on-premise, como o Active Directory, o DNS e o DHCP, estabeleceu a base da autenticação centralizada e do gerenciamento automatizado de endereços e identidades, garantindo maior controle e confiabilidade das conexões locais, além de permitir a auditoria de acessos e a aplicação de políticas de segurança consistentes.

No âmbito em nuvem, a implantação dos serviços de HTTP, FTP e MariaDB possibilitou a hospedagem de aplicações e bancos de dados de forma escalável, acessível e alinhada às demandas de modernização da instituição. As práticas de segurança implementadas, incluindo restrições de acesso via firewall, monitoramento de logs e autenticação de dispositivos, reforçam a proteção dos serviços e a integridade dos dados, complementando a infraestrutura híbrida.

Com a consolidação desses serviços e a aplicação de medidas de segurança, a Universidade Polaris passa a dispor de uma arquitetura de rede estruturada, padronizada e preparada para futuras expansões. O projeto demonstra, portanto, a importância de integrar práticas de gestão de rede, segurança da informação e computação em nuvem na construção de um ecossistema tecnológico acadêmico eficiente, seguro e sustentável.