PONTIFÍCIA UNIVERISDADE CATÓLICA DE GOIÁS

ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO

GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

VIRTUALIZAÇÃO DE SERVIDORES

YANN SANTANA MACIEL DE LIMA

GOIÂNIA

2021

YANN SANTANA MACIEL DE LIMA

VIRTUALIZAÇÃO DE SERVIDORES

Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Orientadora: Profa. Dra. Solange Da Silva

Co-orientador: Prof. Me. Gildenor de Souza A Cavalcante

GOIÂNIA

2021

YANN SANTANA MACIEL DE LIMA

VIRTUALIZAÇÃO DE SERVIDORES

Este Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, em ____/_____/_____.

_____

Profa. Ma. Ludmilla Reis Pinheiro dos Santos

Coordenadora de Trabalho de Conclusão de Curso

Banca examinadora:

_____

Orientadora: Profa. Dra. Solange da Silva

_____

Co-orientador: Prof. Me. Gildenor De Souza A Cavalcante

_____

Prof. Me. Rafael Leal Martins

_____

Prof. Dr. Fabio Barbosa Rodrigues

GOIÂNIA

2021

DEDICATÓRIA

A Deus, que sempre esteve comigo na minha luta para vencer todos os desafios. A minha mãe e meu pai, por serem meus exemplos.

AGRADECIMENTOS

À Deus por tudo.

Aos meus pais pela presença e pelo apoio com carinho, dedicação e por serem minha motivação.

À professora Orientadora Solange da Silva e ao professor co-orientador Gildenor de Souza A Cavalcante, pois foram fundamentais para elaboração e aperfeiçoamento deste trabalho, além de sua dedicação e apoio.

Aos meus colegas por me ajudarem em vários momentos.

A todos, que de uma maneira ou de outra, colaboraram para a minha formação.

RESUMO

Com a pandemia do COVID, percebeu-se a necessidade de técnicas alternativas para implementação de servidores. O objetivo geral deste trabalho é implementar uma rede virtual em nuvem, utilizando a plataforma AWS e verificar se ela proporciona usabilidade, confiabilidade e desempenho. Quanto aos procedimentos técnicos foram utilizadas as pesquisas bibliográfica e experimental. Os resultados obtidos mostram que é possível implementar servidores de forma diferente do tradicional. Concluiu-se que utilizando a plataforma AWS, a virtualização de servidores proporciona usabilidade, confiabilidade e desempenho.

Palavras-chave: Virtualização de servidores. Rede Virtual em Nuvem. Usabilidade. Confiabilidade. Desempenho.

ABSTRACT

With the COVID pandemic, the need for alternative techniques for server implementation was realized. The general objective of this work is to implement a virtual cloud network, using the AWS platform and verify if it provides usability, reliability and performance. As for technical procedures, bibliographic and experimental research were used. The results obtained show that it is possible to implement servers differently from the traditional one. It was concluded that using the AWS platform, server virtualization provides usability, reliability and performance.

Keywords: Server virtualization. Virtual Cloud Network. Usability. Reliability. Performance.

LISTA DE FIGURAS

LISTAS DE SIGLAS

ACL	*Access Control List* ou Lista de Controle de Acesso
AMI	*Amazon Machine Image* ou Imagens de Máquina da Amazon
AWS	*Amazon Web Service* ou Serviços *Web* da Amazon
Bits	*Binary Digit* ou *D*igito Binário
Bytes	*Binary Term ou* Termos Binários
CPF	Cadastro de Pessoa Física
CPU	*Central Process Unit* ou Unidade de Controle de Processamento
DHCP	*Dynamic Host Configuration Protocol* ou Protocolo de Configuração Dinâmica de *Hosts*
DMZ	*Demilitarized Zone* ou Zona Desmilitarizada
DNS	*Domain Name System* ou Sistemas de Nomes de Domínio
EC2	*Elastic Compute Cloud* ou Nuvem de Computação Elástica
FTP	*File Transfer Protocol* ou Protocolo de Transferência de Arquivos
GB	*GigaByte*
Gbps	*Gigabits* por segundo
GHz	*GigaHertz*
HD	*Hard Disk* ou Disco Rígido
HTML	*HyperText Markup Language* ou Linguagem de Marcação de Hipertexto
HTTP	*HyperText Transfer Protocol* ou Protocolo de Transferência de Hypertexto
HTTPS	*Hyper Text Transfer Protocol Secure ou* Protocolo de Transferência de Hypertexto Seguro
IP	*Internet Protocol* ou Protocolo de *Internet*
IPv4	*Internet Protocol version 4* ou Protocolo de *Internet* versão 4
ISO	*International Organization for Standardization ou* Organização Internacional para Padronização
KB	*KiloBytes*
Kbps	*KiloBits* por segundo
LAN	*Local Área Network* ou Rede Local

MB	*MegaBytes*
Mbps	*Megabits* por segundo
PHP	*Hypertext Preprocessor* ou Pré-Processador de Hypertexto
RAM	*Random Access Memory* ou Memória de Acesso Randômico
SO	*Operating System* ou Sistema Operacional
SSD	*Solid-State Drive* ou Unidade de Estado Sólido
TI	Tecnologia da Informação
URL	*Uniform Resource Locator* ou Localizador Uniforme de Recursos
VDI	*Virtual Desktop Infrastructure* ou Infraestrutura de *Desktops* Virtuais
VPC	*Virtual Private Cloud* ou Nuvem Privada Virtual
VPN	*Virtual Private Network* ou Rede Virtual Privada
WLAN	*Virtual Local Área Network* ou Rede Local Virtual
WWW	*World Wide Web* ou Rede Mundial de Computadores

SUMÁRIO

1 INTRODUÇÃO

A virtualização de servidor é uma técnica que visa melhorar a flexibilidade e os recursos computacionais, distribuindo de um único sistema computacional em diversos outros recursos, separando o sistema operacional do *hardware*, como por exemplo servidores e armazenamento. A virtualização de *desktop* possibilita o usuário acessar virtualmente áreas de trabalhos diferentes com sistemas operacionais diferentes, como por exemplo, *Windows* 7 ou *Windows* 10, tendo assim a capacidade de utilizar *Desktops* em um único computador, além de separar um espaço para o usuário e um espaço para a empresa (HUH e SEO, 2016).

Já para Ogunyemi e Johnston (2017), a virtualização de servidores tem como objetivo aumentar a produtividade e otimizar os recursos computacionais dos servidores. Ela possibilita que várias máquinas virtuais possam ser executadas em um computador servidor físico, além disso, elas encapsulam esse servidor fazendo com que ela se comporte como servidores físicos.

No entanto, para Lucena (2016) a virtualização de servidores serve para "dividir os recursos de um *hardware* em diversos servidores virtuais", assim pode-se utilizar vários sistemas operacionais no mesmo *hardware* sem interferência de sistemas.

Amazon Web Service (AWS) é uma plataforma de *cloud muito* usada. Ela oferece mais de 175 serviços de *datacenters* no mundo. Grandes empresas e órgãos governamentais utilizam a AWS para reduzirem custos, obtendo agilidade de inovação (AWS, 2020).

A plataforma AWS disponibiliza serviços como armazenamento, computação, banco de dados, inteligência artificial, *Internet* das Coisas ou *Internet ou Things* (IoT) e *data lakes*. Ela foi criada para ser um dos melhores ambientes de *cloud computing*, além de ser um dos mais flexíveis e seguros do mercado. Ela também possui 77 zonas de disponibilidade, espalhadas em 24 regiões geográficas em todo mundo, no qual uma dessas regiões é em São Paulo (AWS, 2020).

A Amazon *Virtual Private Cloud* (VPC) permite criar rede na Nuvem AWS isolada logicamente de outras redes, podendo controlar totalmente as redes virtuais, incluindo dividir endereços IP, criar sub-redes, configurar tabelas de roteamento e gateways. É possível também utilizar IPv4 e IPv6 nas VPC. Pode-se criar sub-redes

acessíveis ao público e sub-redes privadas acessíveis somente a rede VPC. Também é possível colocar sistema *back-end*, como servidores de aplicativos ou banco de dados nas redes privadas. Por último, é possível criar várias camadas de segurança como *Segurity Group* e ACLs (AWS,2020).

O serviço *Amazon Elastic Compute Cloud* (EC2) é possível criar instâncias escolhendo o sistema operacional, processador, processador gráfico, armazenamento, rede e formas de compra. A AWS disponibiliza os processadores mais velozes na nuvem, com opções de rede de 100 *Gigabits* por segundos (Gbps) (AWS, 2020).

O Amazon *WorkSpaces* é um serviço de *desktop* seguro, gerenciado, rápido e podendo acessar *desktops* em qualquer momento, qualquer lugar e qualquer dispositivo compatível. Ela fornece *desktops Windows* 10 ou Linux Ubuntu 18.04, possibilitando a virtualização de *desktop*. Esse serviço auxilia na eliminação de complexidade no gerenciamento de inventário de *hardwares*, patches e versões de SO e infraestrutura de *desktops* virtuais (VDI) ajudando na simplificação de entrega de *desktops* (AWS, 2020).

Justifica-se estudar este tema, pois as atividades econômicas necessitam de respostas rápidas, eficientes e que contemplem a redução de custos. Além disso, o advento da pandemia do CORONAVIRUS impôs um forte isolamento social. Portanto, existe uma constante busca por tecnologias mais simples, com mais facilidade de acesso e manutenção. Huh e Seo (2016) afirmam que a virtualização de servidores traz benefícios tais como: a redução de custos com gerenciamento, proteção de rack e diminuição de consumo de energia. Seguindo às regras sanitárias de isolamento, várias atividades estão sendo executadas em regime *home office*, utilizando o domínio da empresa, porém caso surja algum problema em seus servidores, os técnicos precisam se deslocar até lá para consertá-lo. Com a virtualização, isso poderia ser evitado.

Diante do contexto, este projeto visa responder a seguinte questão de pesquisa: **- A virtualização de servidores pode proporcionar usabilidade, confiabilidade e desempenho?**

O objetivo geral deste trabalho é **implementar uma rede virtual em nuvem utilizando a plataforma AWS e identificar se ela proporciona usabilidade, confiabilidade e desempenho.**

Os objetivos específicos são:

- Implementar virtualização de:
  a. Implementação da rede VPC, suas sub-redes, *Firewall,* Rede DMZ e servidor *Web Proxy.*
  b. Servidores *WEB*, FTP, DNS e VPN.
  c. Implementação de *desktops* e banco de dados virtualizados.
  d. Testes e análise de usabilidade, confiabilidade e desempenho.

Espera-se que os resultados deste trabalho possam contribuir:

- com a implementação de servidores diferente do tradicional.
- apresentando a usabilidade, confiabilidade e desempenho de uma rede de servidores convencional.

Esta monografia está estruturada da seguinte maneira:

Neste Capítulo é apresentada a introdução com o contexto do trabalho, a questão de pesquisa, objetivos e resultados esperados. O Capítulo 2 traz o referencial teórico com conceitos e definições e trabalhos relacionados com o tema. No Capítulo 3 estão descritos os procedimentos metodológicos, mostrando o que foi feito para atingir o objetivo geral.

No Capítulo 4 está apresentada a criação da rede VPC, domínio e implementação da segurança. Também traz a implementação dos servidores FTP, *Web Proxy*, *Web* e VPN. Apresenta também a implementação dos *desktops* virtualizados utilizando *WorkSpaces*. O Capítulo 5 contém análise dos resultados obtidos, apresentando os testes com os *Firewalls*, servidores e *Workspaces* para analisar a confiabilidade, usabilidade e desempenho. Finalmente, o Capítulo 6 traz a conclusão, apresentando a resposta da questão de pesquisa e sugestões de trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 Conceitos e definições

Para Dias (2017), a área de Tecnologia da Informação (TI) procura estudar atividades e soluções que foram coletadas através de recursos computacionais. Ela tem como objetivo receber, armazenar, gerenciar, acessar e utilizar informações obtidas através de recursos computacionais. Os profissionais da área projetam, implementam e atualizam resoluções computacionais.

Varella (2019) afirma que o Cadastro de Pessoa Física (CPF) é um documento que possui onze dígitos de numeração que tem como objetivo identificar contribuintes. Ele é utilizado para identificar o indivíduo em casos como, por exemplo, quando o mesmo for prestar concursos públicos, realizar abertura de conta nos bancos, solicitar empréstimos, cartões de créditos, financiamento, carteira de trabalho e carteira de transporte.

Para Salutes (2019), *Internet Protocol* (IP) ou Protocolo de *Internet*, funciona como “um CPF de pessoa física”, possibilitando que dispositivos sejam identificados e conexões sejam realizadas a partir de uma sequência de números.

Um pacote ou datagrama são informações para controlar os dados do usuário. Os dados do usuário são roteados entre origem e destino na *internet* ou por qualquer rede que utilize pacotes. Um pacote tem como objetivo ser um *contêiner*, possuindo informações de endereço e destino, detecção de erros, forma de sequenciamento, formas de correção e os dados. Essas informações servem para garantir a entrega dos dados com confiabilidade (SPEEDCHECK, 2020).

Para Clemente (2019) *Host* é todo computador ligado a alguma rede via IP e domínio. Ele oferece recursos, informações e serviços aos usuários. A *Internet* é um exemplo de *host*.

Para Souza (2020) o *ping* estabelece a disponibilidade do *host*, a distância entre os equipamentos de rede e verificam o tempo de resposta de envio e recebimento desses equipamentos. Ele se refere ao tempo que demora para um pacote de dados seja transmitido de um *host* ao servidor na *Internet* e retornar ao *host*.

Para Dias (2018), uma tabela de roteamento é uma tabela que possui registros de destinos para encaminhamentos de pacotes. Então ela informa se o destino está na rede e se estiver, para onde o pacote deve ser encaminhado.

Tabelas de rotas possui um conjunto de regras (rotas), que determinam a direção do tráfego da rede ou do *gateway*. As tabelas de rotas controlam a direção do tráfego de uma rede VPC (AWS,2020).

Para Tripathi e Hubballi (2017), o protocolo *Dynamic Host Configuration Protocol* (DHCP), em português protocolo de configuração dinâmica de *hosts*, é utilizado para a obtenção de parâmetros de configuração de redes como, por exemplo, o endereço de IP de um servidor, no qual o cliente solicita um endereço IP e o servidor DHCP libera um IP para o cliente, em que esse IP possa ser IPv4 ou IPv6 dependendo da configuração escolhida. Já para Montanari (2019), o protocolo DHCP é “um meio para os computadores conseguirem um endereço IP automaticamente”.

Masuda, Segawa e Mori (2019) afirmam que como a comunicação entre computadores é realizada via IP, o protocolo *Domain Name System* (DNS) ou em português, Sistemas de Nomes de Domínio mostra-se importante, pois ele converte os endereços IP para nomes reconhecidos para os humanos. Este autor informa também que o servidor DNS gerencia e opera distribuindo nomes de domínio, traduzindo, por exemplo, um endereço de IP para um endereço de site.

Para Costa (2020), *Firewall* é uma ferramenta de *software* ou *hardware* utilizada para filtrar dados que cheguem na rede, impedindo que dados indesejados trafeguem entrem na sua rede interna garantindo a segurança de sua rede. *Firewall* é dividido em três, filtro de pacotes, proxies e inspeção de dados. O filtro de pacotes utilizando uma Lista de Controle de Acesso (ACL), analisa todos os pacotes que passam pela rede, tanto entrando quanto saindo, permitindo e bloqueando de acordo sua configuração. Os *proxies* mascaram o endereço IP dos seus usuários e filtra as mensagens que eles recebem. E, por último, a inspeção de dados, que é responsável por inspecionar o fluxo de dados de uma ponta a outra na rede, fazendo a inspeção de cabeçalhos e o estado dos pacotes recebidos.

Para Fernandes (2019) *Web Proxy* é um intermediário entre o servidor e o usuário pois todos os dados do usuário devem passar pelo *Web Proxy*, onde ele

atende as requisições e repassa esses dados. Também é possível a inserção de regras para a regras, possibilitando bloqueio de sites para certos usuários.

De acordo com Iskandar, Virma e Ahmar (2018), a *Demilitarized Zone* (DMZ) ou Zona desmilitarizada, é uma camada responsável pela segurança da rede interna, para proteger as portas que são visíveis para computadores do mundo externo. Assim, quando acontecem ataques, o invasor só poderá acessar o *host* DMZ sem ter acesso a rede interna. Também é dito que a principal função do DMZ é que ele controla o tráfego da rede, movendo todos os serviços da rede interna para outra rede. É como se criasse uma área entre as redes interna e externa. Já para Filipe (2019) DMZ é uma "rede entre duas redes". É interessante seu uso, pois ela dá mais segurança por acrescentar mais uma camada dentro do *firewall*.

De acordo com Muxfeldt (2017), *Local Área Network* (LAN), ou Rede local, são vários computadores conectados entre si, pertencentes a uma organização, através de uma rede que tem como função estabelecer uma comunicação de computador para computador sem um servidor central, ou uma comunicação onde possui um servidor central que forneça os serviços de redes aos outros computadores.

Morellato (2018) conceitua *Virtual Local Área Network* (VLAN) ou rede local virtual com o sendo uma técnica que junta máquinas conectadas a redes locais, de maneira lógica. Ele define VLANs como “redes logicamente independentes”. Com isso, podemos dividir a rede local e várias redes virtuais, gerando domínios de broadcast independentes.

Para Reis (2017) Sub-rede é definida como "uma subdivisão lógica de uma rede IP". Esta subdivisão permite diminuir tráfego de rede, aumentar a performance da rede e simplificar seu gerenciamento. Este autor também afirma que uma máscara de Sub-rede tem como função informar qual porção representa uma rede e qual porção representa os *hosts*. A máscara IPv4 é um endereço de IP no qual é dividido em 4 bytes divididos por pontos. A máscara é usada para informar a quantidade de bits do endereço IP são utilizados para identificar a rede e as sub-redes.

Internet Gateway é um componente da rede VPC, um serviço oferecido pela AWS. Ela permite a comunicação entre a *Internet* e a VPC e fornece os destinos nas tabelas de roteamento da VPC para o tráfego na *Internet* e executa a *network address*

translation (NAT), para instâncias que utilizam IPv4 públicos. Ela também tem suporte aos IPv4 e IPv6 (AWS,2020).

Um *security group* é um serviço oferecido pela AWS. Ela controla o tráfego de entrada e saída de uma instância por meio de regras. Ela não atua no nível de rede e sim no nível da instância. Portanto, ela é uma camada a mais de segurança (AWS,2020).

Jianyun e LiChunyan (2018) afirmam que uma rede virtual privada (VPN) é uma conexão temporária e segura que utiliza uma rede pública no qual ela cria um túnel seguro e estável em ambiente caótico como uma rede pública. Ela serve para conectar usuários de forma remota com segurança, como por exemplo, uma conexão de uma matriz de uma empresa e sua filial.

OpenVPN fornece serviços VPN. Ele provê flexibilidade em soluções VPN com objetivo de proteger a comunicação de dados. É possível ser implantado em servidores convencionais, dispositivos virtuais ou em servidores na nuvem (OPENVPN, 2020).

Admin Web UI é a interface *Web* de gerenciamento do servidor *Web*. As configurações salvas utilizando esse artificio não são enviadas ao servidor de imediato pois é necessário atualizar o servidor utilizando um botão que aparece após realizar as configurações (OPENVPN, 2020).

Para Ruwaida e Kurnia (2018), o protocolo de transferência de arquivos (FTP) é um protocolo utilizado para transferências de arquivos de um *host* para outro utilizando a porta 21. Esse protocolo usa arquitetura cliente-servidor. Os usuários se autenticam normalmente por *login* e senha, porém também podem acessar anonimamente o servidor, se este permitir. Esse protocolo é bastante escolhido se for desejado armazenar arquivos ou dados, com velocidade no processo de *download* e *upload* entre o servidor e o cliente.

Pimenta (2020) afirma que *browser*, que significa navegadores de *Internet*, é um *software* que é usado para acessar a *Internet*. Também permite visitar sites, visualizar mídias, enviar e/ou receber *e-mail*, além de outras funcionalidades.

Para Pereira (2014) *World Wide Web* (WWW) ou somente "*web*" é um dos vários meios de acesso à rede *internet*. Ela a fornece serviços tais com: como FTP,

troca de mensagens, *e-mail* e páginas. Este autor também afirma que a *Web* utiliza do protocolo HTTP para enviar e receber informações, sendo dependente de *browsers*.

De acordo com Tavares (2019), os servidores *WEB* são responsáveis por processarem requisições dos usuários, tais como guardar dados ou enviar respostas para as requisições dos clientes. Ele informa que é nestes servidores que estão sendo hospedadas imagens, páginas e códigos de um site de *Internet*.

De acordo com Moraes (2018), *Uniform Resource Locator* (URL) ou Localizador Uniforme de recursos, é o “endereço virtual” de um site permitindo que um site possa ser acessado na rede.

Souza (2019) explica que o protocolo *Hypertext Transfer Protocol* (HTTP) ou protocolo de transferência de hipertexto, serve para os usuários que inserirem a URL de um site possam ter acesso aos dados e conteúdo dele. Este autor afirma que “um navegador é um cliente HTTP”, pois quando um usuário insere uma URL em seu navegador ele cria uma solicitação HTTP na *Internet* e a envia no endereço IP da URL. Desta forma, o servidor recebe a solicitação e envia os dados requisitados para o usuário.

Para Marques (2019), o *HyperText Markup Language* (HTML) é o “componente básico da *web*”, que possibilita estabelecer a estrutura básica de um site e a inserção de conteúdo, ou seja, realiza a organização das informações de um site.

Segundo Felipe (2020), *Hypertext Preprocessor* (PHP) é uma linguagem de programação usada para o desenvolvimento de sites dinâmicos, integração de aplicações e para agilizar o desenvolvimento de sistemas. Este autor informa que essa linguagem pode ser usada para comunicar-se com o servidor (*back-end*) e com cliente (*front-end*), porém é mais utilizada como *back-end*. PHP é usado para desenvolvimento de sites pois pode ser integrado a um arquivo HTML.

Longen (2020) afirma que o *Apache* é um servidor de código aberto para aplicações *Web* e sua função é estabelecer conexões entre o servidor e os navegadores enquanto baixa e entrega arquivos entre eles. A comunicação entre o servidor e o cliente ocorre mediante o protocolo HTTP. O servidor Apache atua facilitando e assegurando a comunicação entre eles.

Para Pimenta (2020), Sistema Operacional (SO) é o principal *software* de um computador. Ele gerencia todos os outros *softwares* e *hardware* do computador, além de gerenciar memória e os processos do computador. Ele também permite que os usuários possam se comunicar com o computador sem que precisar saber seu idioma.

De acordo com Longen (2020), *FileZilla* é um *software* de transferência de arquivos. Ele consegue transferir grandes arquivos com segurança e velocidade. Utiliza protocolo FTP ou protocolo PHP para transferência de arquivos, porem existe limitações quando se trata de *upload* utilizando PHP, limitação esta que não existe usando FTP. *FileZilla* possui interface de fácil entendimento e bastante intuitivo e possui a possibilidade de substituição de arquivos.

De acordo com Macêdo (2016), *Wireshark* é "um analisador de protocolos de forma gráfica", que serve para auxiliar na análise de cada pacote que passa por uma rede, ou seja, ele captura o tráfego de rede. O *Wireshark* serve também para decodificar vários protocolos.

Para Lando (2019) *International Organization for Standardization* (ISO) ou Organização Internacional para Padronização é uma organização que possui objetivo de "desenvolver e promover normas que possam ser utilizadas por todos os países do mundo". Foi fundada em 1946 e é sediada na cidade de Genebra, na Suiça. Atua concedendo certificação para organizações que possuem um sistema de gestão de qualidade que esteja de acordo com as normas ISO.

Volpato (2016) afirma que a ISO 9241-11 define que usabilidade é uma norma de qualidade de *software*, na qual ela é nível de eficácia, eficiência e satisfação em que os usuários conseguem atingir seus objetivos. Usabilidade é aplicada em objetos como sites, tela, *smartphone*, computador, óculos, aplicativo ou qualquer dispositivo.

Ruggieri (2016), em um resumo que fez da ISO 9126, afirma que a confiabilidade é uma característica de qualidade de *software*. Ela dita um grupo de atributos que mostram a capacidade do *software* em manter seu desempenho nas condições que foram estabelecidas durante uma porção de tempo estabelecida.

Segundo Gomes (2015), desempenho para computação é uma medida de qualidade que tem como principal proposito determinar a velocidade de execução, qual o mínimo de desempenho aceitável. Além disso, mostra quanto é consumido do

processador e da memória e quantos usuários conseguem executar de forma concorrente.

MD5 é um algoritmo criptográfico que possui o principal objetivo de verificar se um arquivo não foi alterado. Ele produz uma soma de verificação em ambos os conjuntos, e então, compara as somas de verificação para confirmar se são as mesmas (TI-FORENSE, 2018).

Segundo Alencar (2020), *Google Chrome* é um dos navegadores mais populares, disponível para todos os SO, com exceção do *Windows Phone*. É um navegador desenvolvido pelo *Google*, possuindo versão em português além de ser gratuito.

Para Costa (2020), *Firefox* é um navegador de código aberto utilizado para acessar a sites com segurança, com boa experiencia de usuário e com possibilidade de instalação de extensões.

Souza (2020) afirma que banco de dados "é a organização e armazenagem de informações". Ou seja, é o agrupamento de dados relacionados que necessitam ser armazenados para segurança ou conferência futura. O banco de dados melhora armazenar, acessar e recuperar dados.

De acordo com Longen (2019), o *MySQL* é um banco de dados relacional, que utiliza o modelo cliente-servidor, sendo um *software* de código aberto para criação e gerenciamento de banco de dados relacionais.

O Amazon *Relational Database Service* (RDS) é o serviço que facilita a operação, configuração e a escalabilidade de bancos de dados relacionais na nuvem, além de otimizar memoria, performance e entrada e saída de dados. Oferece bancos de dados comuns como *MySQL*, MariaDB e *PostgreSQL* (AWS,2021).

Para Sales (2019), os Testes de Cargas avaliam a capacidade da aplicação de manter um certo nível especificado de qualidade de desempenho, diante de uma grande quantidade de dados.

Gusti (2019) afirma que o aplicativo Apache JMeter é um *software* livre, usado para realizar testes de carga. É um aplicativo grátis, que pode ser executado em várias plataformas com dispositivos *desktop,* sendo possível simular vários usuários, com encadeamentos paralelos, para criar cargas pesadas em aplicativos *web*.

2.2 Trabalhos Relacionados

Existem trabalhos relacionados na literatura, tais como Huh e Seo (2016), que realizaram um trabalho experimental, implementando virtualização de servidores, assim como Ogunyemi e Johnston (2017), que entrevistaram empresas que implementaram a virtualização de servidores para analisar se valia a pena ou não. Já Ghannoum e Rodrigues (2018) fizeram um estudo no tema de virtualização de servidores para verificar as vantagens e as desvantagens de utilizar essa técnica e Fernandes e Nuno (2018), que substituíram os seus servidores físicos para servidores virtualizados e coletaram os benefícios dessa mudança.

O trabalho de Huh e Seo (2016) teve como objetivo encontrar uma alternativa para diminuir o custo com consumo de energia, devido aos servidores convencionais consumirem bastante, e por esses servidores entrarem em desuso após 5 anos. Assim, ele propõe a utilização da virtualização logica e física para a resolução do problema. Eles construíram o servidor da seguinte forma: criaram o servidor DHCP, o servidor DNS, o servidor *Web*. Instalou os *softwares VMware Work station* 7.1.0 e o *VMware vShere* Cliente 4.1.0 nos clientes e iInstalou o *software VMware* ESXi 4.1.0 no servidor *Web*, DNS e de *e-mails*. Esses servidores estavam separados em LANs diferentes, porém foram conectados por VLANs e foram combinados em um único hub. Para avaliar o desempenho ele utilizou método *Wireshark*. Por fim, ele verificou que o sistema de virtualização de servidor projetado pode ser implementado por um custo bem menor que na forma convencional. Eles fizeram o experimento, mas não compararam com nenhum outro servidor e nem relataram quais foram os resultados, apenas mostraram o passo-a-passo de como fazer a virtualização dos servidores.

Ogunyemi e Johnston (2017) não implementaram, mas pesquisaram se a implementação de virtualização de servidores é um bom investimento para empresas ou organizações públicas. Eles buscavam dar uma alternativa às organizações, principalmente as que estão em países em desenvolvimento, sobre a viabilidade de implementar a virtualização de servidores. Eles entrevistaram 83 organizações que atuavam em setores, tais como: TI, varejo, telecomunicações, bancos e finanças, manufatura, petróleo, gás, energia, serviços governamentais, desenvolvedores de *software* e serviços de *Internet* como amostras para esse estudo. Os entrevistados foram executivos da área de TI dessas organizações.

Ogunyemi e Johnston (2017) Observaram que as organizações trabalhavam, observavam quais os *softwares* que eles usaram para implementar a virtualização. Verificaram se a implementação foi limitada ou extensiva. Contaram quantos servidores essas organizações possuíam, os seus orçamentos. Perguntaram quais foram os benefícios e as desvantagens do uso da virtualização de servidores e quais foram os fatores que inibem ou permitem sua implementação. Daí concluíram que estas organizações que utilizaram a virtualização de servidores indicaram tiveram um bom resultado na sua implementação, porém tiveram como desvantagens problemas de gerenciamento e falta de habilidades de TI.

Ghannoum e Rodrigues (2018) fizeram um trabalho visando estudar os diferentes temas de virtualização. Definiram o que é Servidor, Máquina Virtual, *Hypervisor*. Informaram que existem virtualização de aplicativos, de *desktops* e de servidores. São estes os três tipos de virtualização. Mostraram que a técnica de virtualização de servidores proporciona baixo custo, maior segurança (na virtualização total de servidores), facilidade de gerenciamento, compatibilidade, otimização da utilização do *hardware* e a possibilidade de configuração de ambientes de testes. Porém, como desvantagem, essa técnica tende a proporcionar pouco espaço de disco, falta de acesso direto ao *hardware*, pouca segurança, se não for bem tratada, maquinas ociosas devido ao acumulo de maquinas virtuais. Existe a necessidade de ter servidores físicos e alguns aplicativos ou os sistemas podem perder performance. Concluíram que a virtualização de servidores pode ser vantajosa, porém é necessário um estudo prévio de seu ambiente para verificar se é a melhor opção.

Fernandes e Nuno (2018) substituíram os servidores físicos que possuíam por servidores virtualizados e verificaram o benefício de utilizar a virtualização. Eles possuíam 9 servidores físicos e 2 racks. Para virtualizar, fizeram a virtualização dos 9 servidores físicos, criaram um servidor de domínio para contingencia e criaram 2 servidores físicos com o SO *VMWare* ESXi 5.0. Após a virtualização dos servidores, eles possuíam apenas 2 servidores físicos, 1 rack e 10 servidores virtuais. Concluíram que com isto reduziram custos e consumo de energia, redundância nos controladores de domínio, devido a criação do servidor de Domínio para contingencia. Também foi concluído que a virtualização de servidores otimiza o uso do hardware, além do ganho de espaço físico, diminuindo o custo do ar condicionado. Por fim, devido a facilidade

de aplicar políticas para recuperação de quedas, a virtualização proporcionou facilidade no gerenciamento, maior segurança e durabilidade.

3 PROCEDIMENTOS METODOLOGICOS

A natureza dessa pesquisa é um resumo de assunto, pois busca sistematizar a área de virtualização de servidores. O resumo de assunto procura sistematizar uma área de conhecimento, no qual indica sua evolução histórica e seu estado da arte, ou seja, adequado para os cursos de graduação (WAZLAWICK, 2014).

Segundos seus objetivos é uma pesquisa descritiva, pois busca alcançar dados consistentes sobre virtualização de servidores, porém sem obter teorias que esclareçam esse fenômeno (WAZLAWICK, 2014).

Segundo os procedimentos técnicos esta pesquisa é bibliográfica e experimental, pois pretende fazer um levantamento teórico da virtualização de servidores e implementar esta técnica. Wazlawick (2014) sugere que a pesquisa bibliográfica deve seguir os seguintes passos:

a) Listar periódicos e eventos relevantes ao tema da pesquisa e periódicos gerais sobre computação para verificar se existe algum artigo na área do tema a ser pesquisado.
b) Pesquisar artigos publicados a pelo menos cinco anos atrás.
c) Selecionar da lista, títulos relacionados ao tema a ser pesquisado.
d) Ler o *abstract* dos artigos selecionados e classificá-los em relevância "alta", “média” ou "baixa".
e) Ler artigos com alta relevância e fazer resumos com os principais assuntos aprendidos sobre o tema. Anotar títulos que podem ser mencionados na bibliográfica mesmo que tenham mais de cinco anos.
f) Se necessário, ler artigos de relevância média ou baixa, porém sempre priorizar os artigos com alta relevância.
g) Aluno decide se já tem possui material suficiente para elaborar uma pesquisa consistente.
h) Se precisar aumentar a pesquisa lendo artigos mais antigos (Expandindo o passo *b*) ou periódicos com grau de relevância menor (Expandindo o passo *a*).

Para Wazlawick (2014), pesquisa experimental se dá pela manipulação de uma parte da realidade do pesquisador, como por exemplo, esse trabalho, que introduz uma nova técnica para verificar se vale ou não a pena utilizar virtualização de

servidores. Na pesquisa experimental é necessário possuir variáveis manipuláveis pelo pesquisador e variáveis de observação. A medição dessa variável de observação que pode concluir se existe alguma dependência entre ela e alguma variável manipulável. No caso desse trabalho, as variáveis manipuláveis são os componentes da rede, ou seja, os servidores, *desktops* e funções para segurança. As variáveis observadas são a usabilidade, confiabilidade e desempenho.

Para Gil (2002), uma pesquisa experimental depende de um objeto de estudo, variáveis que podem manipulá-lo, formas de controle das variáveis e formas de observação dos efeitos que as variáveis produzem no objeto.

Gil (2002) define que para a realização de uma pesquisa experimental é necessário seguir os seguintes passos:

A) O problema dessa pesquisa é: = Virtualização de servidores poderia proporcionar usabilidade, confiabilidade e desempenho do servidor físico?

B) Quanto a definição do plano experimental, inicialmente todos os servidores foram estudados e depois eles serão implementados, conforme item c, abaixo.

C) Quanto ao ambiente experimental foi implementado na plataforma AWS, da seguinte forma:

- Servidor *Firewall* foi implementado utilizando o recurso ACL oferecido pela plataforma AWS.
- Servidor DNS foi implementado no Serviço *Route53,* oferecido pela plataforma AWS em conjunto com o serviço de criação de nomes de domínio do site *Freenom*.
- As *Security Groups* foram implementadas em cada instância (exceção dos clientes *Workspaces)* realizando papel de rede DMZ.
- A Rede VPC foi dividida - cujo endereçamento interno é 10.0.0.0/26 - em 3 sub-redes: a sub-rede *Workspace*, a sub-rede pública e a sub-rede privada. As sub-redes tem os seguintes endereçamentos:
    a) Sub-rede *Workspace*: 10.0.0.0/28, possui conexão com a rede VPC (10.0.0.0/26) e com a *Internet*.
    b) Sub-rede pública: 10.0.0.16/28, possui conexão com a rede VPC (10.0.0.0/26) e com a *Internet*.

c) Sub-rede privada: 10.0.0.32/28, possui conexão somente com a rede VPC (10.0.0.0/26).

- Servidor FTP foi implementado em uma máquina virtual Microsoft *Windows Server* 2016 *Base*, com configuração de 1 CPU com 2.5GHz, 1GB de memória RAM e Disco Rígido (HD) de 30 GB de espaço, performance de *Internet* de baixa para moderada no padrão AWS (cerca de 100 Mbps de entrada e saída).
- Servidor VPN foi implementado em uma máquina virtual *OpenVPN Access Server,* disponível na AWS, com SO Ubuntu 18LTS, com configuração de 1 CPU com 2.5GHz, 1GB de memória RAM e SSD de 8GB de espaço, performance de *Internet* de baixa para moderada no padrão AWS (cerca de 100Mbps de entrada e saída).
- O Servidor *Web* foi implementado em uma máquina virtual Ubuntu 18.04 LTS - *Bionic*, com configuração de 1 CPU com 2.5GHz, 1GB de memória RAM e SSD de 8 GB de espaço, performance de *Internet* de baixa para moderada no padrão AWS (cerca de 100 Mbps de entrada e saída). Foi utilizado o aplicativo Apache, MySQL e PHP para implementação do servidor *Web* e o navegador *Firefox 86.0 (64 bits).*
- O Servidor *Web Proxy* foi implementado em uma máquina virtual Ubuntu 18.04 LTS - *Bionic*, com configuração de 1 CPU com 2.5GHz, 1GB de memória RAM e SSD de 8 GB de espaço, performance de *Internet* de baixa para moderada no padrão AWS (cerca de 100 Mbps de entrada e saída). Foi usado o aplicativo *Squid* para implementação do servidor *Web Proxy* e navegador *Google Chrome 87.0.4280.88 (64 bits)*.
- Para Virtualização de *Desktops* foi utilizado o serviço *Workspaces* oferecido pela plataforma AWS. Para acessar os *Desktops* foi utilizado o aplicativo *WorkSpaces* versão 3.1.6. Esses *Desktops* usaram SO *Windows* 10 com configuração com 2 CPU, 4 GB de memória RAM e SSD de 80 GB de espaço no qual 50 GB é para uso do usuário. Performance de rede não informada.

D) Para coleta de dados, foi verificado se os servidores estão funcionando e segue o passo a passo para realizar os testes e obter os resultados que

responderão esta questão: - **A virtualização de servidores pode proporcionar confiabilidade, usabilidade e desempenho?**

- **Para testar a confiabilidade**:
    - Na ACL pública foi verificado se logados no servidor VPN ou no *WorkSpaces,* tendo acesso aos sites do Facebook, Twitch, Amazon Prime Video, Instagram e Baixaki. Para esses testes foi utilizado o navegador *Google Chrome*.
    - Na ACL Privada foi verificado se os servidores da sub-rede privada são acessíveis somente por instâncias pertencentes a rede VPC. Assim, foram realizados testes de conexão com os servidores FTP e *Web*, estando logado e deslogado do servidor VPN.
    - O servidor *Web Proxy* foi testado*,* verificando se ele estava bloqueando o acesso aos sites dos nomes cadastrados, utilizando o navegador *Google Chrome*.
    - Foi verificado se as redes DMZs de cada instância estavam permitindo e bloqueando o tráfego corretamente.
    - Foi realizado teste de carga utilizando o aplicativo Apache JMeter, versão 5.4, para verificar se a ACL pública e o *Web Proxy* conseguiam aguentar grande quantidade de solicitações de dados sobre sites que deveriam ser bloqueados.
- **Para verificar a Usabilidade**:
    - Foi verificado o funcionamento do servidor DNS - analisando se os IP's dos servidores haviam sido traduzidos para os *hostnames* escolhidos quando fossem realizados os testes nos servidores.
    - Para o teste de funcionalidade do servidor VPN, foi realizada comunicação com o servidor VPN, utilizando o aplicativo *OpenVPN*, versão 3.2.1(1180). Dentro da rede VPC foi verificado se havia acesso aos servidores FTP e *Web*.
    - Para verificar o funcionamento do servidor FTP, foi observado se clientes VPN conseguiam enviar, receber, excluir ou renomear arquivos que estavam no servidor FTP. O teste foi realizado utilizando o aplicativo *FileZilla* versão 3.50.0.

    - O servidor *Web Proxy* foi verificado, testando se os usuários cadastrados conseguiam se autenticar e observava qual era o IP do cliente (teste foi realizado utilizando o site MeuIP).
    - Foi verificado o servidor *Web*, testando se os clientes VPN conseguiam se conectar com o servidor FTP. Este teste foi realizado com o navegador *Firefox*.
    - Para verificar a funcionalidade do *WorkSpaces*, foi realizada tentativas de conexão com o *desktop* virtual, via aplicativo *WorkSpace,* versão 3.1.6, sendo testado se havia acesso ao servidor FTP e *Web*.
- Para testar o Desempenho, foi verificada a velocidade de *download* e *upload.* O teste foi realizado utilizando o site *Speedtest* nos clientes ao logarem no servidor VPN, nos *desktops Workspaces* e estando deslogado de ambos. Foi verificada a taxa de *download* e *upload* de arquivos, utilizando o *FileZilla* no servidor FTP, logado no servidor VPN e logado no *Workspaces*.

E) Como foi realizada a análise dos resultados:

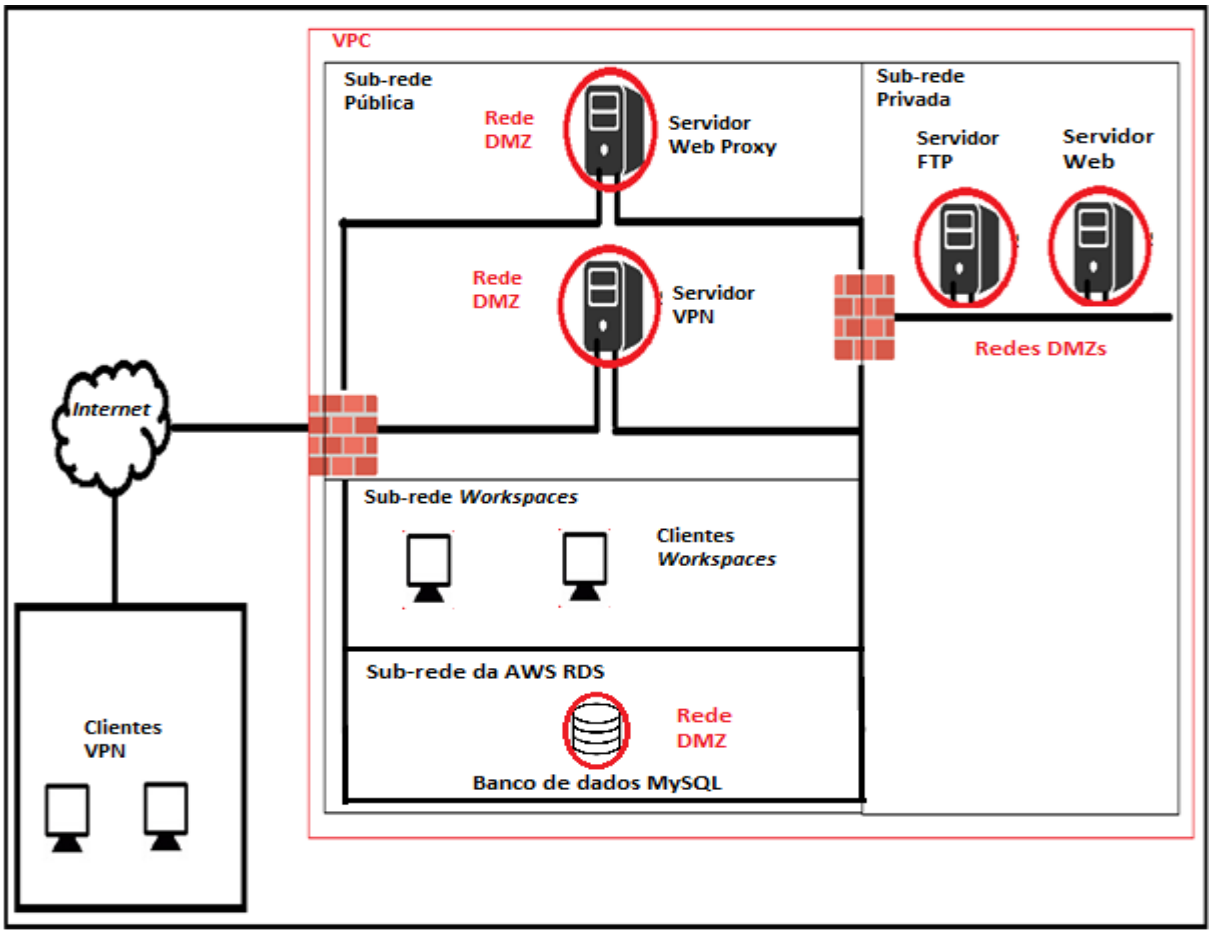
- Foi analisada a confiabilidade sobre os resultados de testes nas ACLs*,* servidor *Web Proxy* e rede DMZ, para verificar se estavam permitindo e ou proibindo o tráfego corretamente, conforme programado. Por fim, foi analisado o comportamento da ACL pública quando ela recebia uma grande quantidade de dados.
- Foi analisada a usabilidade verificando se nos testes, era possível enviar, baixar, excluir e renomear arquivos do servidor FTP. Testou se o servidor *Web* conseguia redirecionar a página *Web* para a página do servidor FTP. E também se os clientes VPN e *Workspaces* conseguiam acessar os recursos do servidor FTP e *Web,* e se os *hostnames,* criados no servidor DNS, estavam sendo traduzidos corretamente.
- Foi analisado o desempenho, verificando a diferença de *download e upload* dos clientes VPN, *Workspaces* e clientes deslogados em ambos. Foi comparada a taxa de *download* e *upload* de arquivos do servidor FTP entre clientes VPN ou *Workspaces*.

F) O trabalho realizado foi registrado em forma de uma monografia de TCC.

4 VIRTUALIZAÇÃO DE SERVIDORES

Nesse trabalho foi implementada uma rede VPC utilizando a plataforma AWS. Conforme mostrado na Figura 1, esta rede possui três sub-redes, dois *firewalls*, quatro servidores, sendo eles, servidor *Web,* servidor *Web Proxy,* servidor VPN e servidor FTP, dois clientes VPN, duas estações *desktop Workspaces* e um banco de dados.

Figura 1 – Diagrama da rede



Fonte: autoria própria

Observa-se que cada instância (exceção dos clientes *Workspaces)* possui sua própria rede DMZ, aumentando ainda mais a sua segurança. Outro ponto a ser considerado é que a sub-rede pública e *Workspaces* possuem acesso a sub-rede privada e a *Internet*, porém a sub-rede privada não possui acesso a *Internet*.

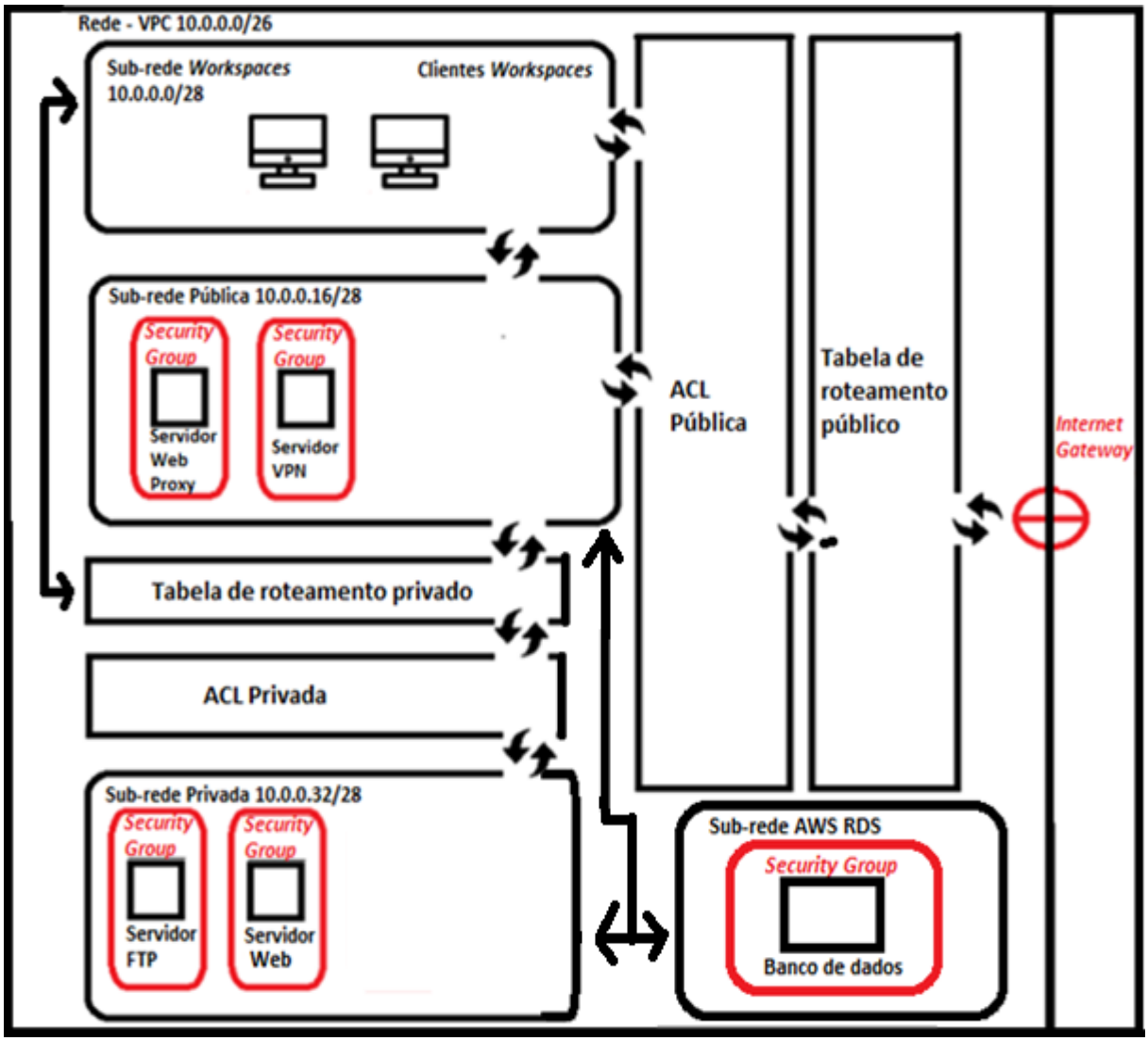
Na Figura 1 é mostrado que para os clientes VPN acessarem os servidores localizados na sub-rede privada, é necessário que o tráfego passe pelo *firewall* público, pela rede DMZ do servidor VPN, pelo *firewall* da sub-rede privada e por fim pela rede DMZ do servidor alvo.

4.1 Rede

A rede foi dividida em três sub-redes com dois *firewalls*. O primeiro *firewall* serve para proteger a sub-rede pública e a sub-rede *Workspaces* e o segundo *firewall* serve para proteger a sub-rede privada.

Conforme apresentado na Figura 2, foram implementadas duas tabelas de roteamento: um roteamento público e um roteamento privado. A tabela de roteamento público é responsável por criar rotas de passagem de pacotes da *Internet* - utilizando *Internet Gateway* - para a sub-rede pública e rotas de passagem de pacotes entre toda a rede VPC. Já a tabela de roteamento privado é responsável por criar rotas de passagem de pacotes entre a rede VPC.

Figura 2 – Diagrama de rede detalhado



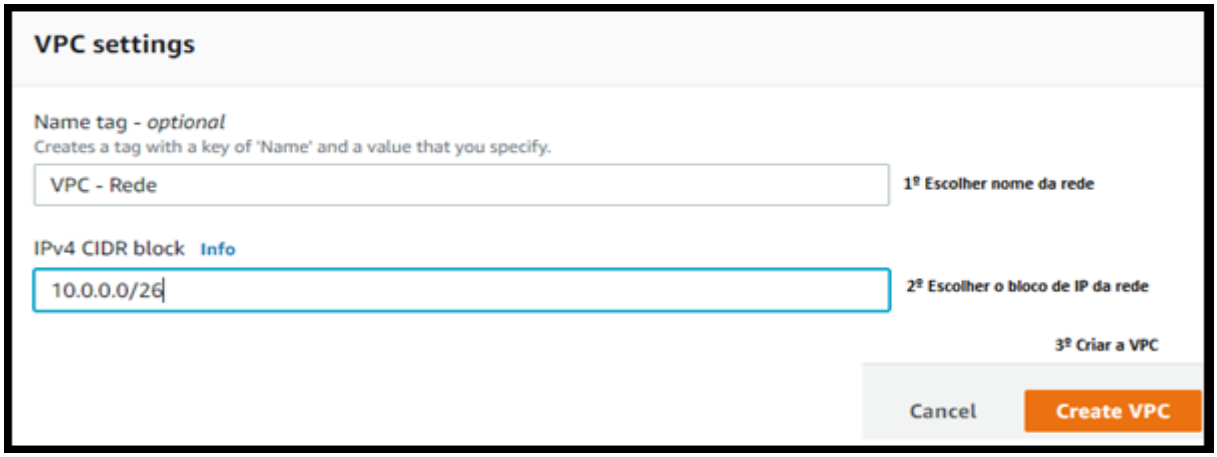
Fonte: Autoria própria

4.1.1 VPC

Para criação da rede VPC, foi escolhida a região de São Paulo para criar a VPC pois tem o menor *ping* devido ela ser a região mais próxima de Goiás.

No serviço VPC da AWS, a rede foi criada conforme ilustra a Figura 3. O nome da rede foi escolhido como sendo “VPC – Rede” e seu bloco de IP foi escolhido como 10.0.0.0/26 que nos proporciona uma faixa de IP disponível de 10.0.0.1 até 10.0.0.62. Foi decidido utilizar poucos IP pois foi implementado poucos servidores e só tem duas máquinas para clientes VPN e duas máquinas para clientes *Workspaces*, e quanto menos IP vagos melhor a segurança.

Figura 3 – Rede VPC



Fonte: alterado baseado em AWS (2020)

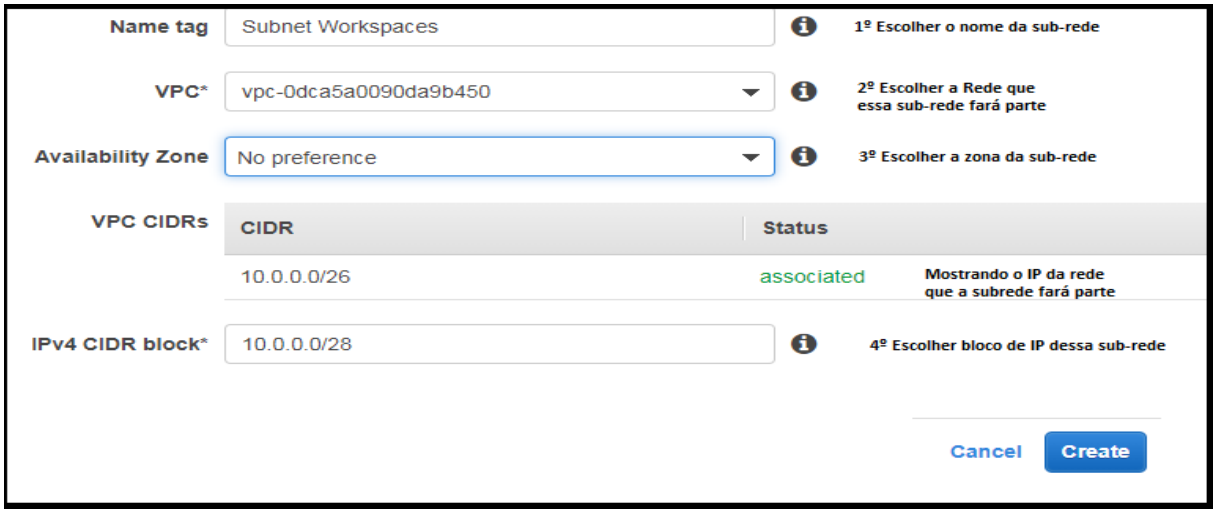
Para que as instâncias associadas a essa VPC acessem funções por nomes ao invés de IP, foi habilitada a opção para que as instâncias recebam nomes de *Host* DNS.

4.1.2 Sub-redes

A rede foi dividida em três sub-redes. A primeira sub-rede foi criada conforme apresentado na Figura 4. Essa sub-rede foi nomeada como “Subnet *Workspaces”*. Foi selecionada a rede “VPC-Rede” para que ela seja uma sub-rede dessa rede. Foi deixado *Default* a zona de disponibilidade. E por último foi escolhido o bloco de IP 10.0.0.0/28, ou seja, é possível adicionar onze *hosts* – AWS reserva o último endereço IP e os quatro primeiros endereços IP da sub-rede para fins de rede IP (AWS, 2020)

- possíveis nessa sub-rede. Os IPs são de 10.0.0.4 até 10.0.0.14 no qual receberam os *desktops Workspaces.*

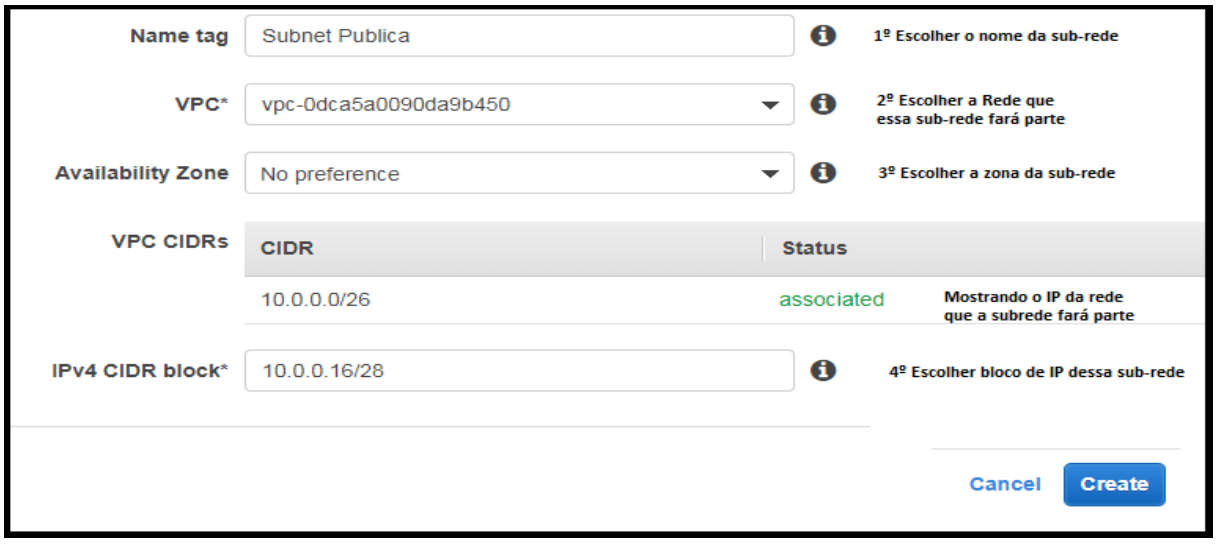
Figura 4 – Sub-rede *Workspaces*



Fonte: alterado baseado em AWS (2020)

A segunda sub-rede foi criada conforme mostrado na Figura 5. Essa sub-rede foi nomeada como “Subnet Publica*”*. Foi selecionada a rede “VPC-Rede” para que ela seja uma sub-rede dessa rede. Foi deixado *Default* a zona de disponibilidade. E por último foi escolhido o bloco de IP 10.0.0.16/28, ou seja, é possível adicionar onze *hosts* possíveis nessa sub-rede. Os IPs são de 10.0.0.19 até 10.0.0.30, no qual receberam o servidor VPN e o servidor *Web Proxy.*

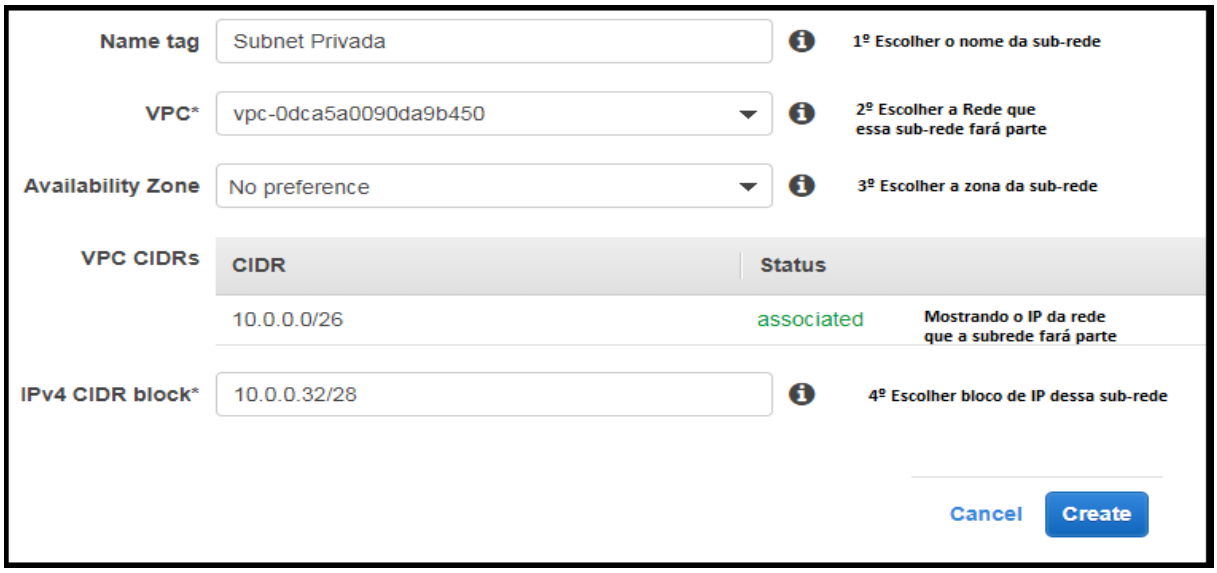
Figura 5 – Sub-rede Pública



Fonte: alterado baseado em AWS (2020)

A terceira sub-rede foi criada ilustrado Figura 6. Essa sub-rede foi nomeada como “Subnet Privada”. Foi selecionada a rede “VPC-Rede” para que ela seja uma sub-rede dessa rede. Foi deixado *Default* a zona de disponibilidade. E por último foi escolhido o bloco de IP 10.0.0.32/28, ou seja, é possível adicionar onze *hosts* possíveis nessa sub-rede. Os IPs são de 10.0.0.35 até 10.0.0.46, no qual receberam os servidores Web e FTP.

Figura 6 – Sub-rede Privada



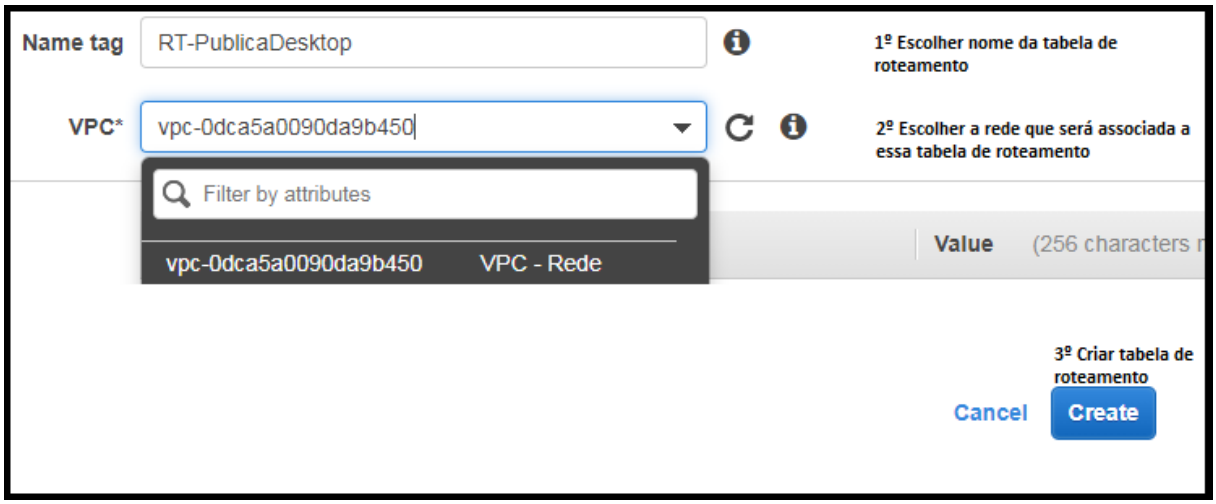
Fonte: alterado baseado em AWS (2020)

Para que as instâncias que estão na sub-rede pública ou sub-rede *Workspaces* 1 e 2 recebam endereço IP público, foi habilitada a auto-atribuição de IP público para instâncias que foram associadas a essas redes.

4.1.3 Tabelas de roteamento e *Internet Gateway*

Foram criadas duas de tabelas de roteamento, uma para a sub-rede *Workspaces* e sub-rede Pública, e uma para a sub-rede privada. Conforme ilustrado na Figura 7, foi criada a tabela de roteamento público e ela foi nomeada como “RT-PublicaDesktop” e foi associada a rede “VPC-rede”. Ao criar a tabela de roteamento, ela, por padrão, já insere uma rota para tráfego entre a rede VPC e isso não foi alterado.

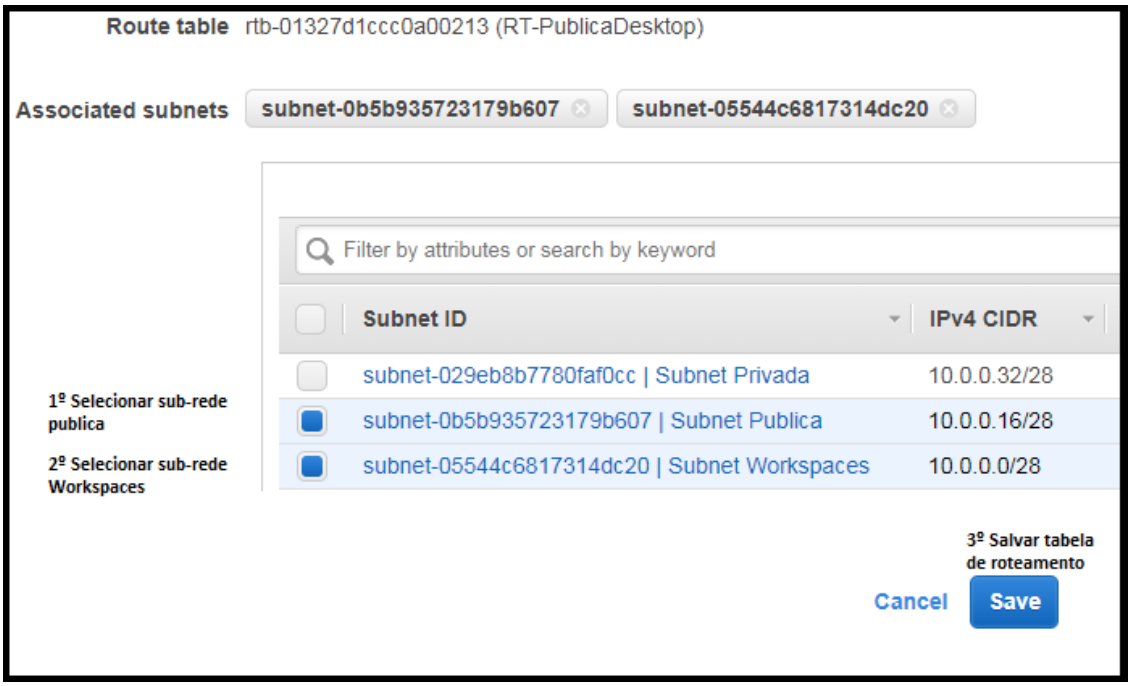
Figura 7 – Criação de tabela de rotas pública



Fonte: alterado baseado em AWS (2020)

Conforme apresentado na Figura 8, após a criação da tabela de rotas pública, foram associadas as sub-redes correspondentes (sub-rede Pública e sub-rede *Workspaces*).

Figura 8 – Associação de sub-redes públicas



Fonte: alterado baseado em AWS (2020)

Foi criado um componente que conecte a nossa sub-rede pública e sub-rede *Workspaces* com a *Internet*. Esse componente é o *Internet Gateway* no qual foi nomeado como “Ig – Rede”.

Após a criação do *Internet Gateway*, foi associado a rede “VPC – Rede” e por fim, conforme apresentado na Figura 9, foi inserido IP de rota padrão. Para que seja possível comunicação com a *Internet* foi selecionada a opção *Internet Gateway* e, então, foi selecionado o *Internet Gateway* “Ig – Rede”, que já havia sido criado. Assim, foi criada uma rota entre a sub-rede pública e sub-rede *Workspaces* com a *Internet*.

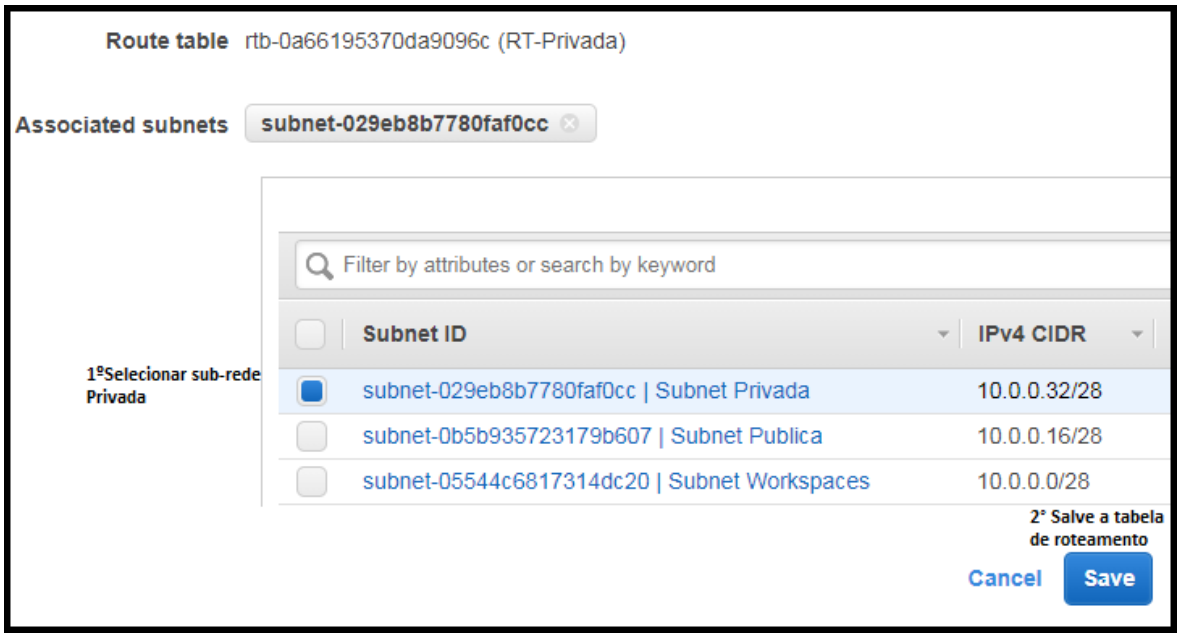
Figura 9 – *Internet gateway* na tabela de rotas

Fonte: alterado baseado em AWS (2020)

A tabela de rotas privadas foi criada conforme mostrado na Figura 7 da página 44, alterando somente o nome da tabela de roteamento essa tabela, no qual foi escolhido o nome “RT-Privada”. Foi associada a rede “VPC-Rede” e então criada. Como dito anteriormente, ao criar uma tabela de roteamento, por padrão, ela já insere uma rota entre a VPC e isso não foi alterado.

Por fim, conforme mostrado na Figura 10, foi selecionada a sub-rede privada para essa tabela de roteamentos. Assim, ela só terá acesso à rede VPC.

Figura 10 - Associação de sub-redes privadas

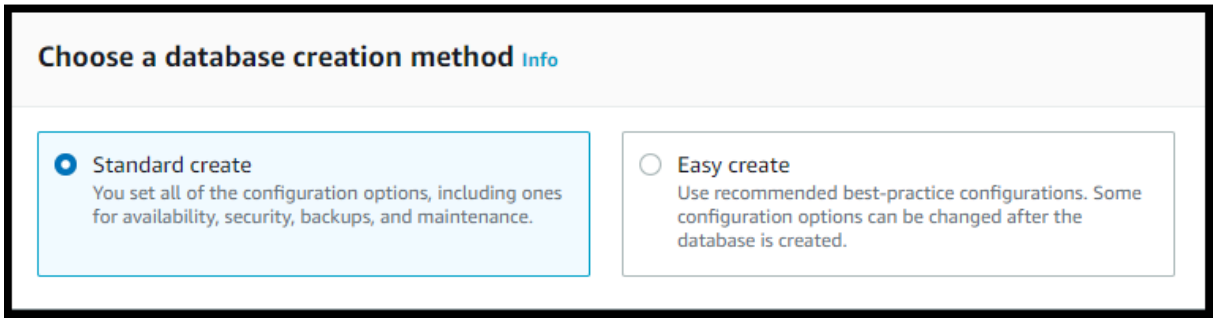


Fonte: alterado baseado em AWS (2020)

4.1.4 Banco de dados RDS

Para armazenamento dos dados de *login* do servidor FTP, foi criado um banco de dados *MySQL*. Para a criação do banco de dados RDS, foi acessado a funcionalidade "*Amazon RDS*", no site da AWS. Nessa página foi selecionada a opção "*Create database*". Como ilustrado na Figura 11, foi escolhido o método "*Standard create"*.

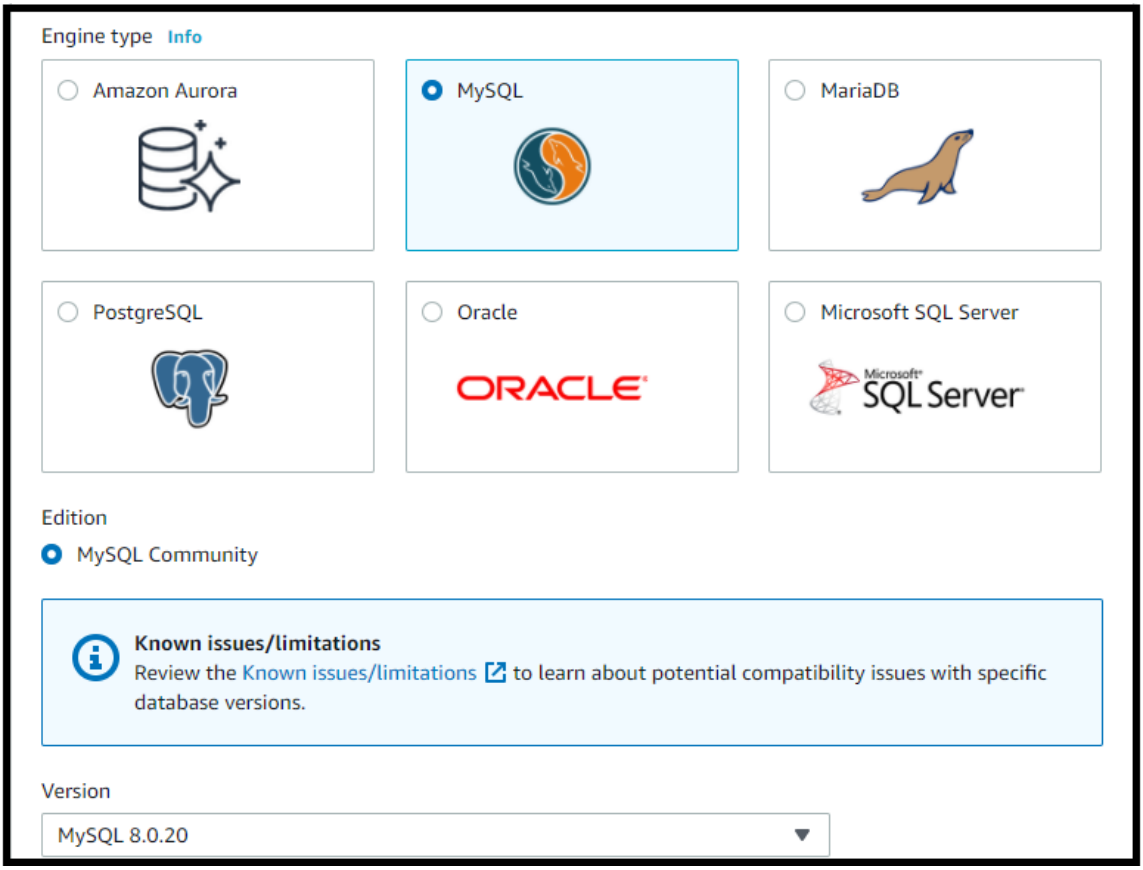
Figura 11 – Método de criação do banco de dados



Fonte: alterado baseado em AWS (2021)

Conforme ilustrado na Figura 12, foi escolhido o *MySQL* como o tipo do banco de dados, assim como a edição "*MySQL Community"* e a versão *MySQL* 8.0.20.

Figura 12 – Escolha do tipo de banco de dados



Fonte: alterado baseado em AWS (2021)

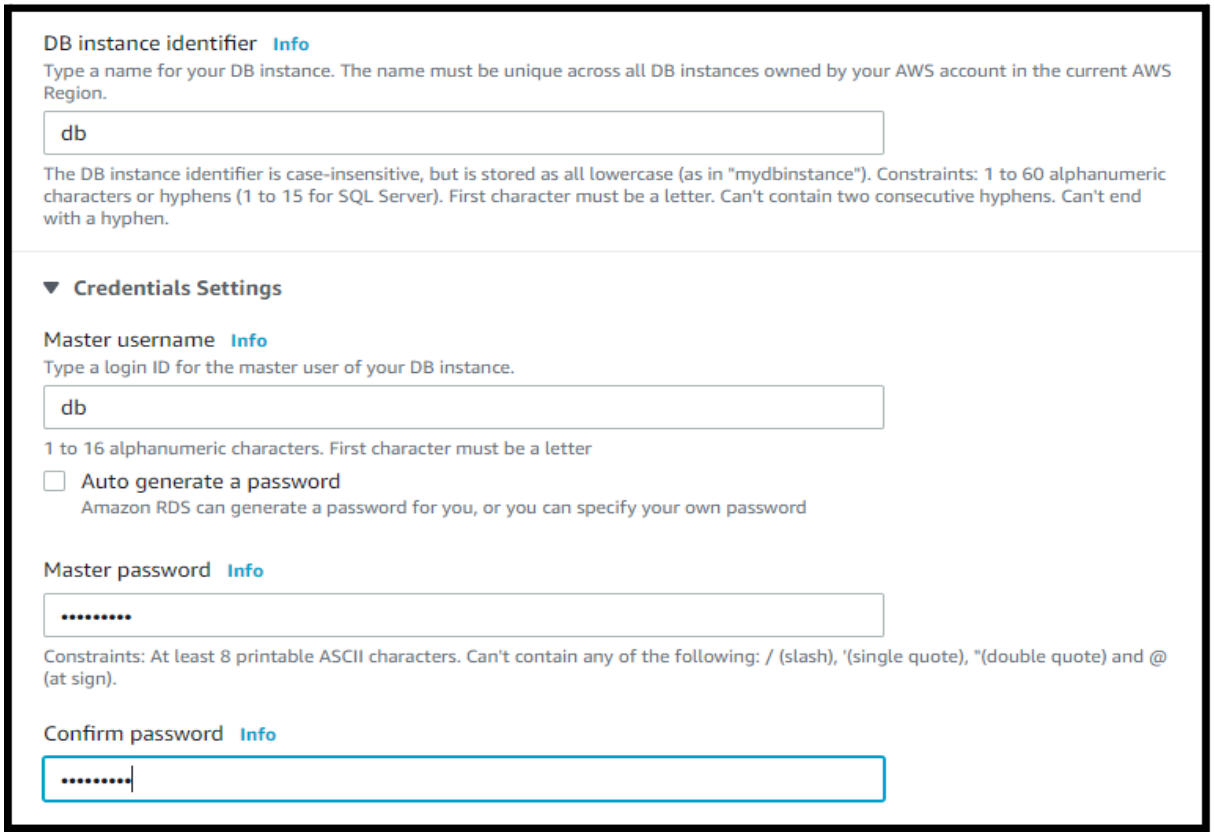
Foi escolhido o *Free Tier* como modelo do banco de dados, como apresentado na Figura 13.

Figura 13 – Escolha do modelo do banco de dados

Fonte: alterado baseado em AWS (2021)

Conforme ilustrado na Figura 14, foi dado o nome “db” para a instância e para o usuário *master* do banco de dados, além de informar a senha e confirmar essa senha.

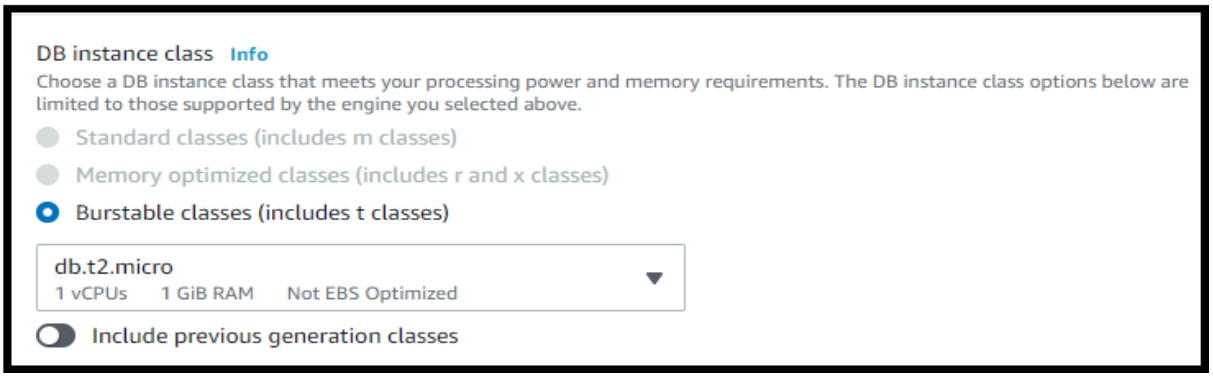
Figura 14 – Configurações da instância do banco de dados



Fonte: alterado baseado em AWS (2021)

A classe de instância foi deixada como *default,* pois essa é a classe do modelo *Free tier,* conforme mostrado na Figura 15.

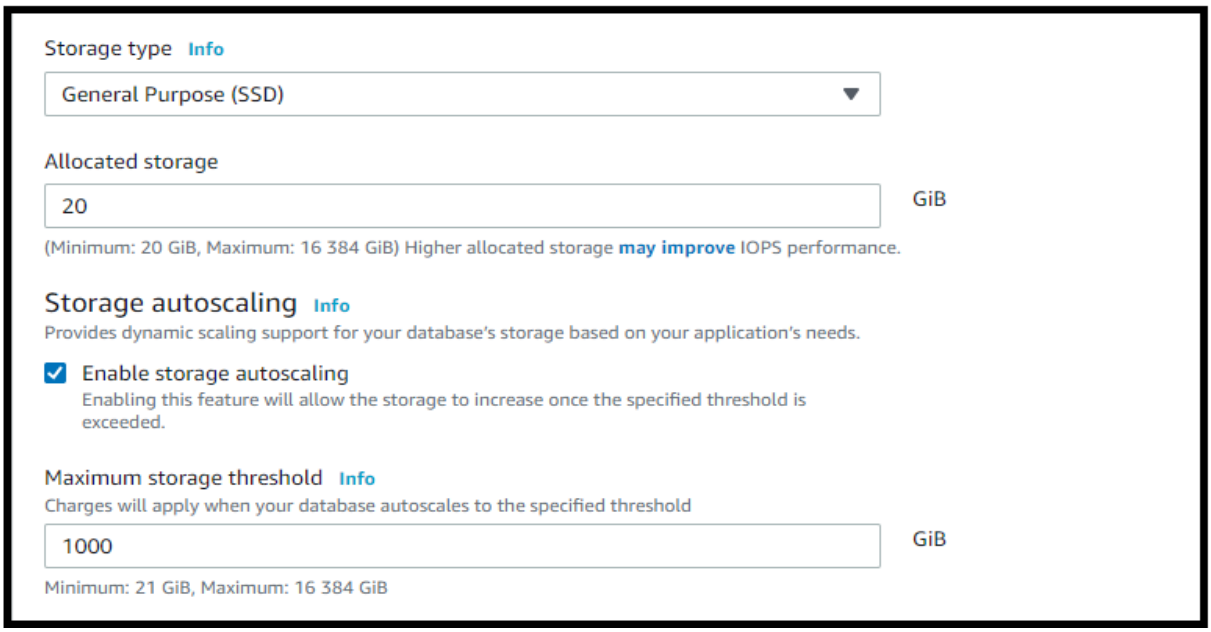
Figura 15 – Classe de instância do banco de dados



Fonte: alterado baseado em AWS (2021)

Como mostrado na Figura 16, o armazenamento também foi deixado *default* devido a esse armazenamento ser solicitado para continuar utilizando o modelo *Free tier*.

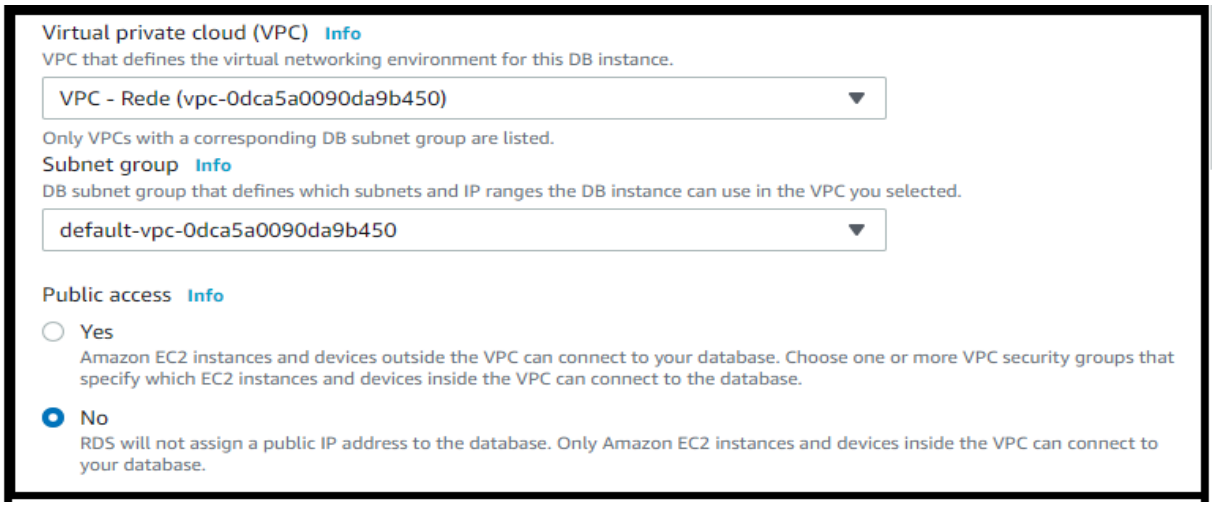
Figura 16 – Armazenamento na criação do banco de dados



Fonte: alterado baseado em AWS (2021)

Como ilustrado na Figura 17, foi escolhido que esse banco de dados pertencesse à “VPC-Rede”, que a sub-rede fosse *default*, não sendo possível acessar esse banco de dados de fora da rede “VPC-Rede”.

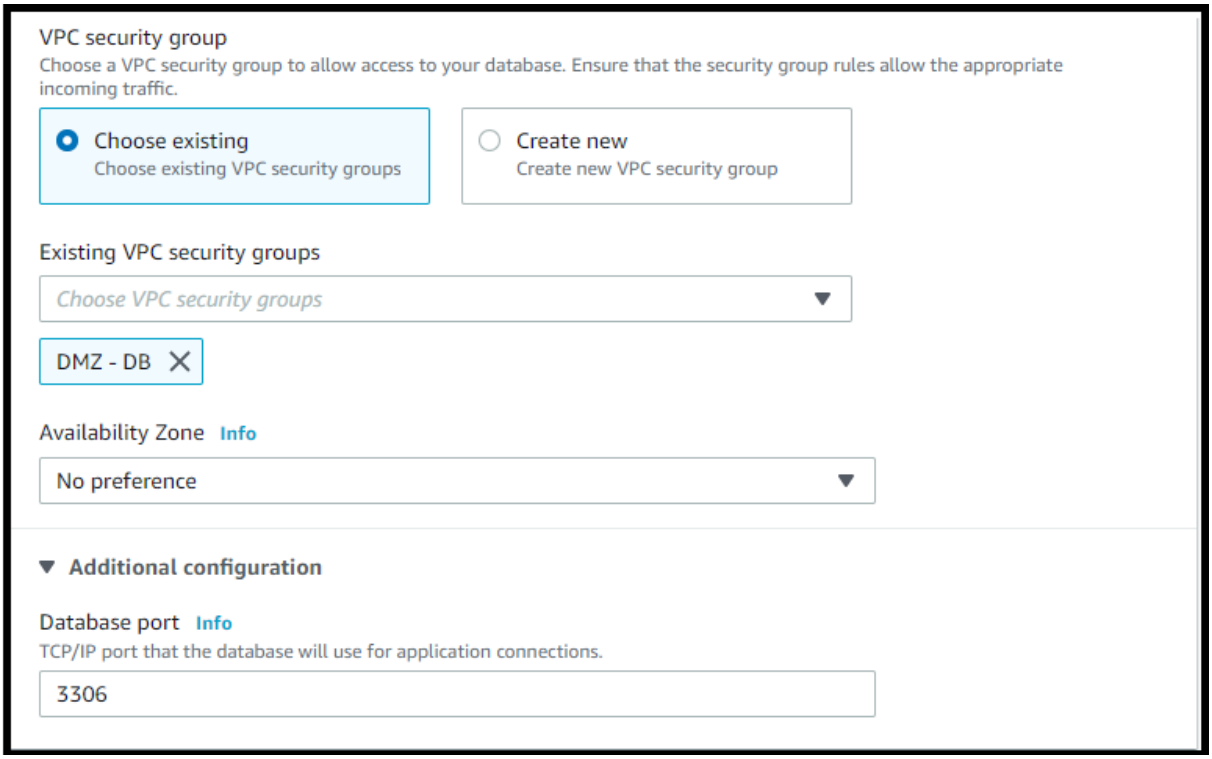
Figura 17 - Escolha da rede no banco de dados



Fonte: alterado baseado em AWS (2021)

Como mostrado na Figura 18, foi escolhido a *security group* “DMZ-DB”, não tendo preferencias na *Availability Zone* e aberta a porta 3306 para comunicação.

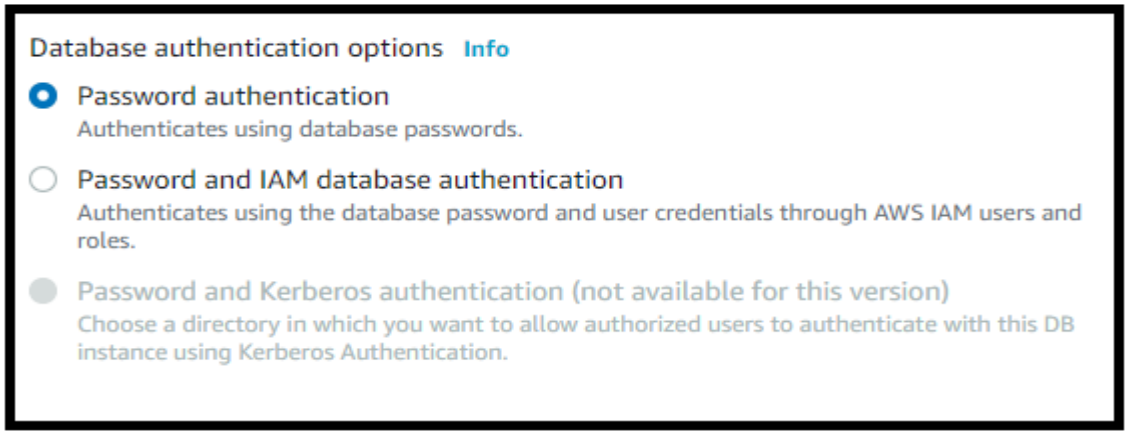
Figura 18 – Escolha da rede DMZ na criação do Banco de dados



Fonte: alterado baseado em AWS (2021)

Foi escolhido que esse banco de dados permitisse a autenticação somente com senha, como mostrado na Figura 19.

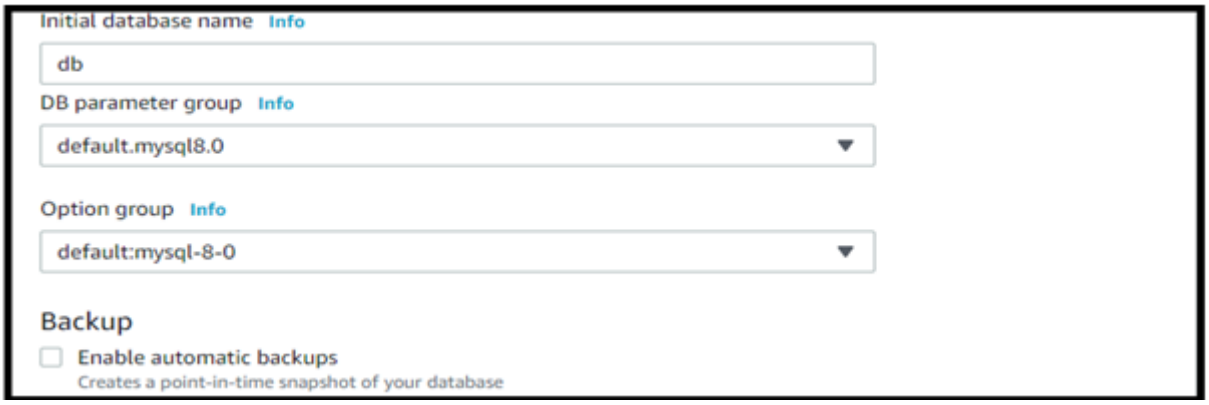
Figura 19 – Modo de autenticação do banco de dados



Fonte: alterado baseado em AWS (2021)

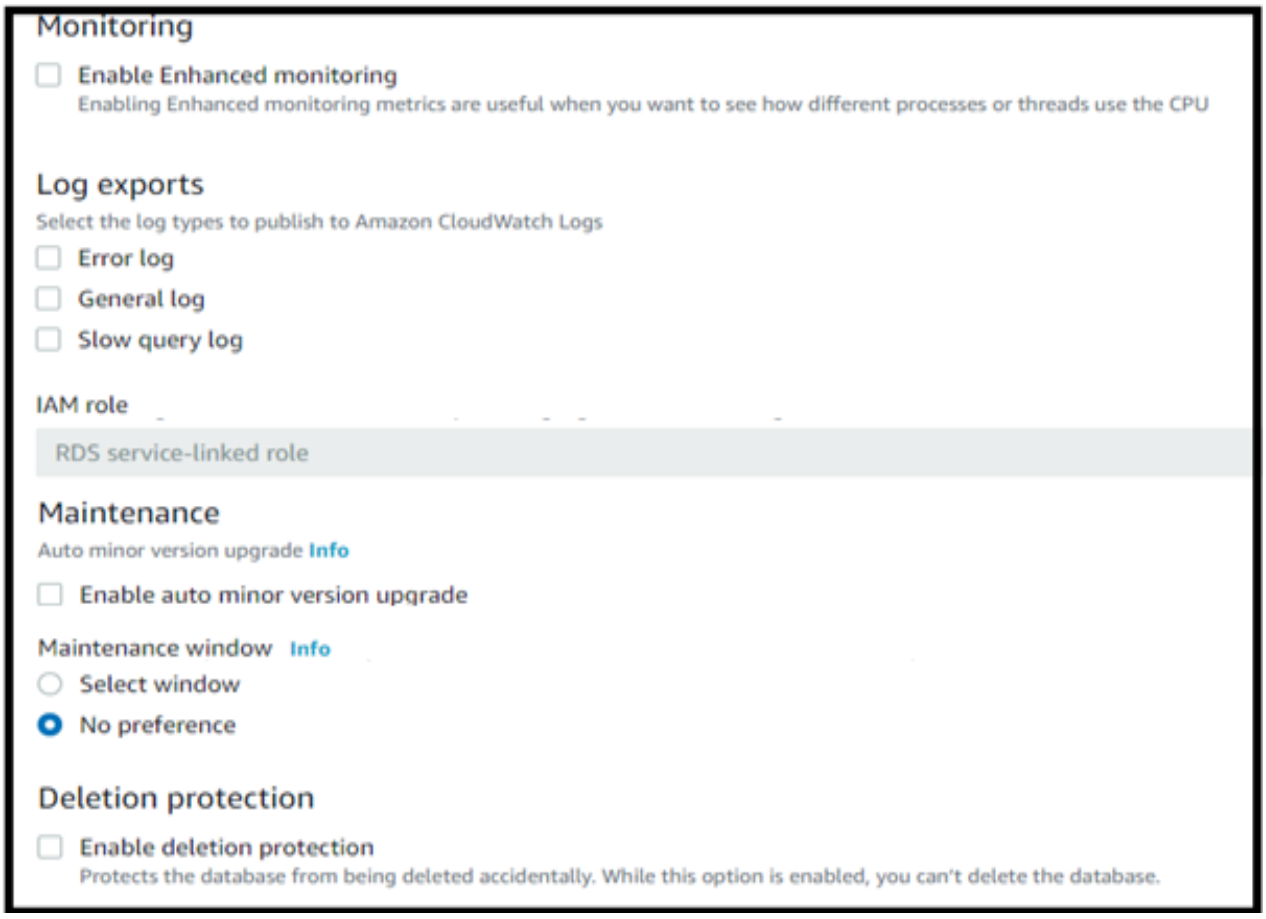
Conforme mostrado na Figura 20 e 21, o nome inicial do banco de dados era “db”, sendo definido que este banco não faria *backup* automático e que as configurações seguintes seriam deixadas como *default,* devido ao *Free tier*.

Figura 20 – Configurações adicionais na criação banco de dados 1



Fonte: alterado baseado em AWS (2021)

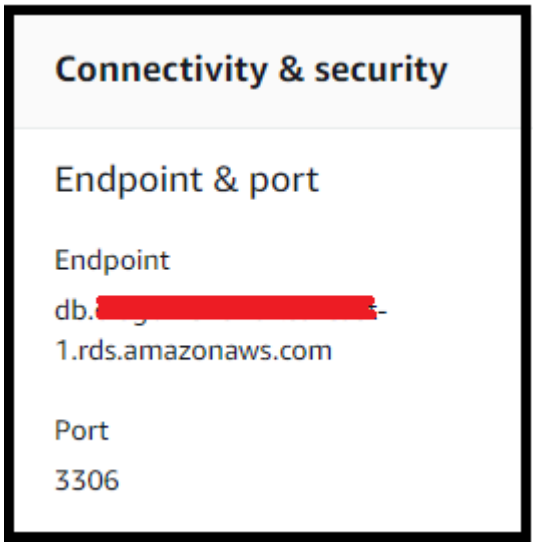
Figura 21 – Configurações adicionais na criação banco de dados 2



Fonte: alterado baseado em AWS (2021)

Por fim, foi selecionada a opção “*Create database*” e, então, o banco de dados foi criado. Depois de criado, armazenou-se o “*end-point*”, que é o *host* desse banco de dados. O *End-point* é apresentado ao clicar em cima do nome desse banco, como ilustrado na Figura 22.

Figura 22 – Localização do *end-point*



Fonte: alterado baseado em AWS (2021)

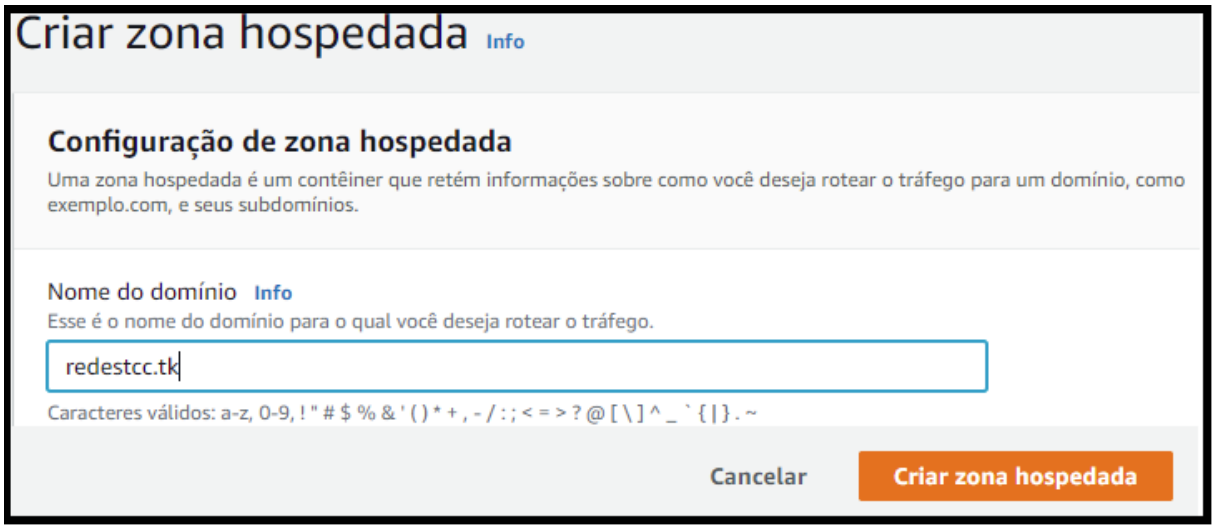
4.1.5 *Route53*

O objetivo do *Route*53 nesse trabalho era de somente redirecionar o tráfego de um nome DNS para um endereço IPv4. Para isso, foi criado uma conta no site *Freenom* para criar um nome de domínio, chamado de “redestcc.tk”. Esse site foi escolhido devido a sua gratuidade em criar um domínio.

Foi acessado a opção “*Route*53” no site da AWS, depois clicou-se na aba “Zonas hospedadas” e por fim no botão “Criar zona hospedada”.

A seguir, conforme mostrado na Figura 23, foi inserido o nome de domínio criado no *Freenom* e, então, foi persistido a opção “Criar zona hospedada”.

Figura 23 – Criação de zona hospedada



Fonte: alterado baseado em AWS (2021)

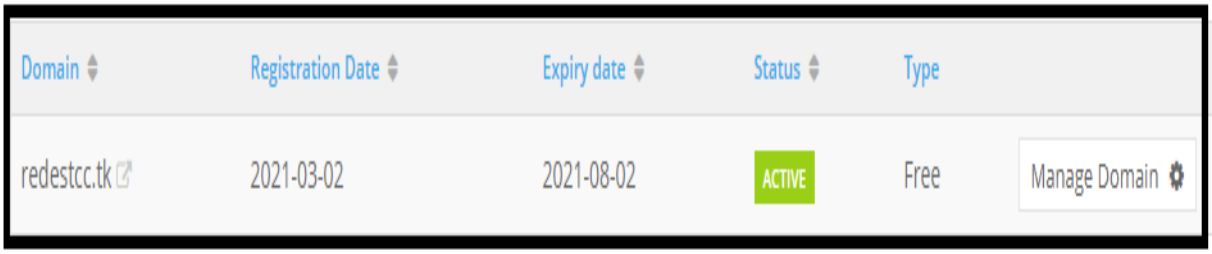
Acessando a zona criada, foram apresentados os registros ilustrados na Figura 24. Esses registros foram utilizados para indicar para onde o domínio aponta e essa configuração é feita no *Freenom*.

Figura 24 – Registros de zona hospedada

Fonte: alterado baseado em AWS (2021)

Foi acessado o site *Freenom* e acessou a conta criada em conjunto com o nome de domínio. No site foi acessado a aba "*Services*" e escolhido a opção "*My Domains*". Após essa ação, foram apresentadas as informações sobre o domínio criado, conforme ilustrado na Figura 25.

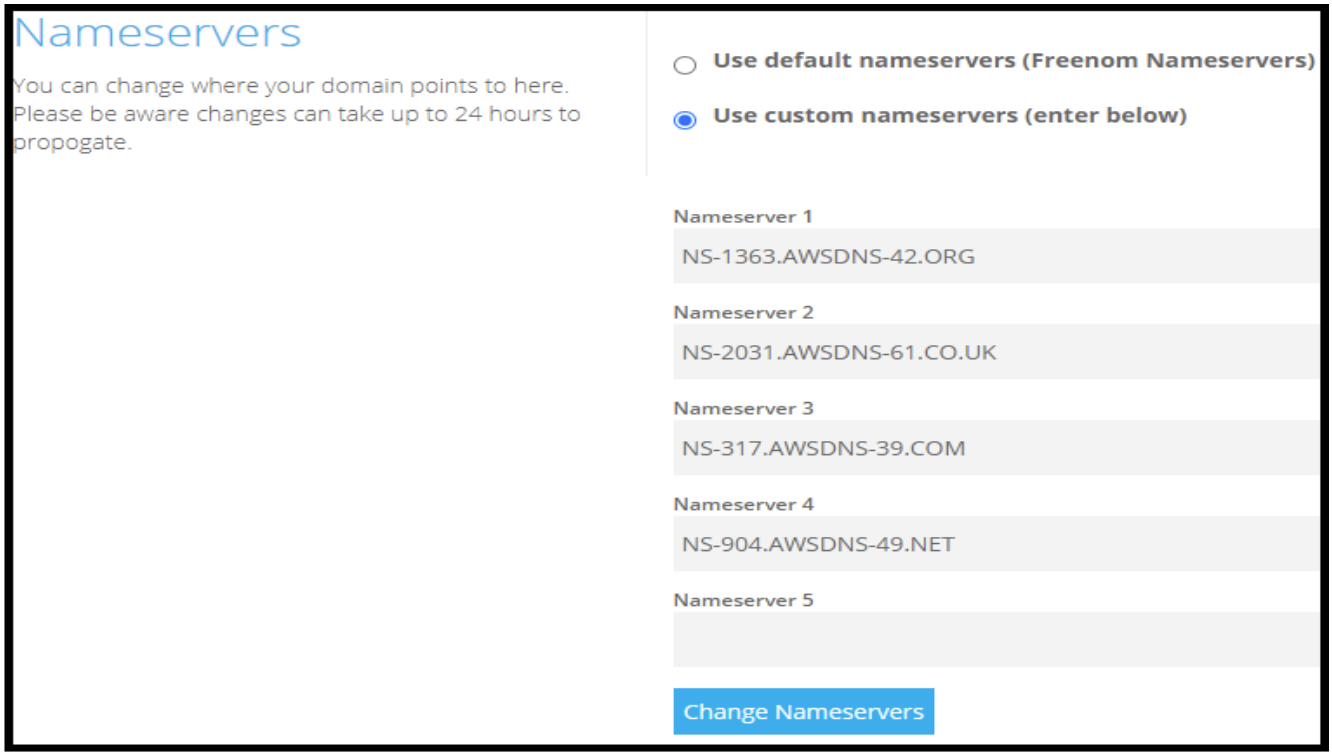
Figura 25 – Nome do domínio no *Freenom*



Fonte: alterado baseado em Freenom (2021)

Depois foi acessada a opção "*Manage Domain*", em seguida a aba "*Management Tools*" e, por fim, a opção "*Nameservers*". Conforme apresentado na Figura 26, foi marcada a opção "*Use custom nameservers (enter below)*". Preenchidas as opções de "*Nameserver*" com as informações contidas na Figura 24 da página 53 e, então, acessada a opção "*Change Nameservers*". Estas alterações podem demorar 24 horas.

Figura 26 – *Nameserver Freenom*

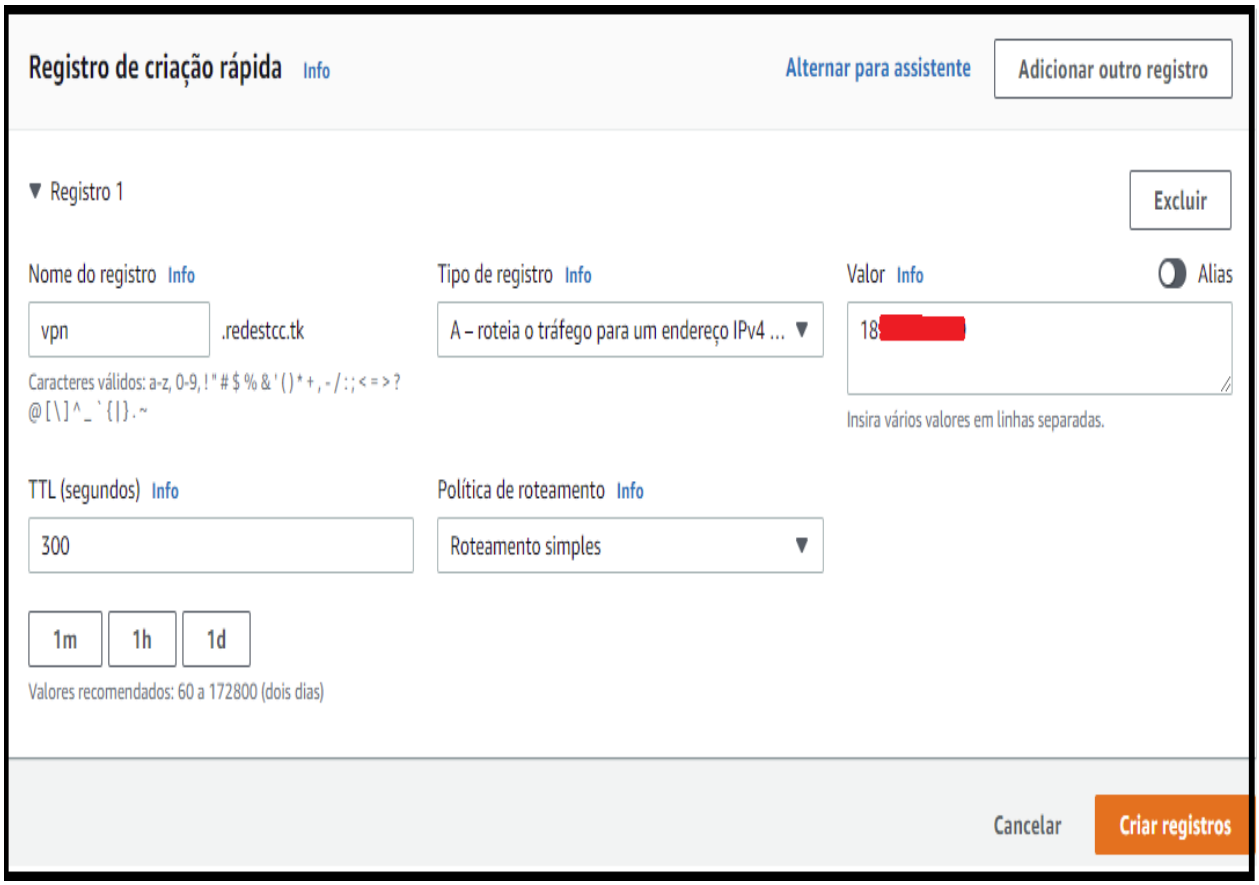


Fonte: alterado baseado em Freenom (2021)

Na página da zona hospedada do domínio criado na AWS (Ilustrado na Figura 24, da página 54), foi selecionada a opção "Criar registro".

É nessa opção que foi especificado qual *hostname* será roteado para o IP do servidor. Na Figura 27 é apresentado o exemplo de criação do registro do servidor VPN, no qual o nome do registro escolhido foi “vpn”, tipo de registro como “A” e o valor como o IP público do servidor VPN. As demais opções foram deixadas como a AWS sugeriu. Logo, o endereço gerado foi “vpn.redestcc.tk”.

Figura 27 – Criação de registro



Fonte: alterado baseado em AWS (2021)

A criação dos registros do servidor Web, Web *Proxy* e FTP, foram feitos conforme mostrados na Figura 27, alterando somente o nome do registro e o IP do servidor. No caso dos servidores FTP e *Web*, como eles não possuem IP público, o IP informado foi seu IP privado.

Portanto, os nomes criados foram “vpn.redestcc.tk” para o servidor VPN, “webproxy.redestcc.tk” para o servidor *Web Proxy*, “web.redestcc.tk” para o servidor *Web* e “ftp.redestcc.tk” para o servidor FTP.

O banco de dados foi registrado conforme ilustrado na Figura 28. O tipo de registro foi escolhido como “*CNAME*”, o nome do registro como “db” e foi inserido na opção valor o endereço *end-point*. Portanto, o nome criado foi “db.redestcc.tk”.

Figura 28 – Criação de registro do banco de dados

Fonte: alterado baseado em AWS (2021)

4.2 Segurança

Conforme apresentado na Figura 1 da página 40, a rede possui dois *firewalls*, um servidor *Web Proxy* e uma rede DMZ para cada instância (exceção dos clientes *Workspaces)*.

Ainda conforme ilustração da Figura 1 da página 40, os clientes *Workspaces* possuem rotas para *Internet*, a sub-rede pública e a sub-rede privada. Para acesso à *Internet*, o tráfego dos clientes *Workspaces* precisam passar pelo *firewall* público e pelo servidor *Web Proxy*. Para acesso a sub-rede pública, o tráfego dos clientes *Workspaces* precisam passar pelo *firewall* público, pelo servidor *Web Proxy* e pela rede DMZ da instância alvo. Para acesso a sub-rede privada, o tráfego precisa passar pelo *firewall* público, servidor *Web Proxy, firewall* da sub-rede privada e pela rede DMZ do servidor alvo.

Conforme mostrado na Figura 1 da página 40, o servidor VPN e o servidor *Web Proxy* possuem rotas para a *Internet*, a sub-rede privada e a sub-rede *Workspaces.* Para acesso à *Internet* e a sub-rede *Workspaces*, o tráfego dos clientes VPC e do

servidor *Web Proxy* precisam passar por suas próprias redes DMZ e pelo *firewall* público. Para acesso a sub-rede privada, o tráfego dos clientes VPN e do servidor *Web Proxy* precisam passar por suas próprias redes DMZ, pelo *firewall* público, pelo *firewall* da sub-rede privada, e pela rede DMZ do servidor alvo.

Por último, conforme ilustrado Figura 1 da página 40, os servidores da sub-rede privada possuem rotas somente para a rede VPC, então ela não possui acesso à *Internet*. Para acesso a sub-rede pública, o tráfego dos servidores da sub-rede privada precisa passar por suas próprias redes DMZ, pelo *firewall* da sub-rede privada, pelo *firewall* público e pela rede DMZ da instância alvo. Para acesso a sub-rede *Workspaces*, o tráfego dos servidores da sub-rede privada precisa passar por suas próprias redes DMZ, pelo *firewall* da sub-rede privada e pelo *firewall* público.

4.2.1 ACL

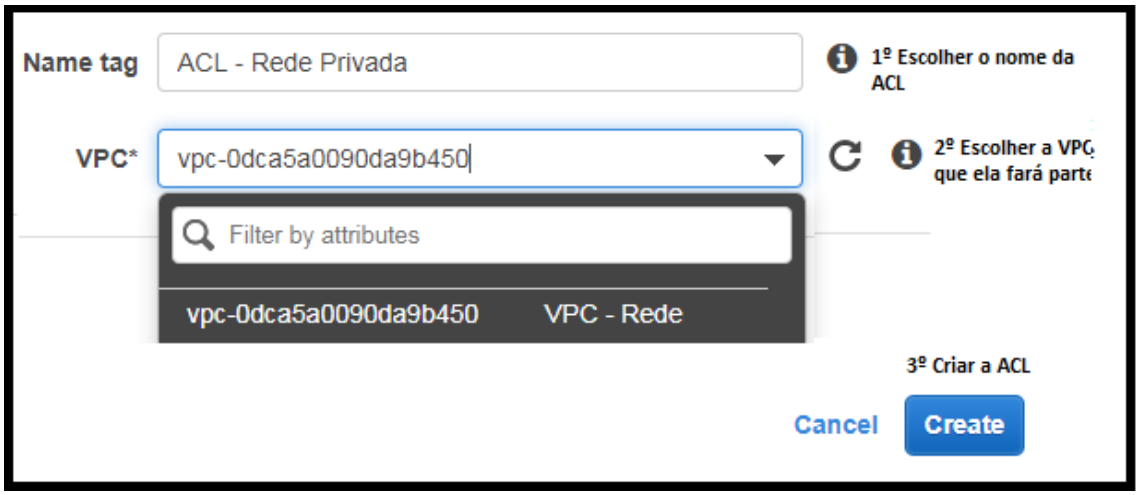
Foram criados dois *firewalls*, o primeiro *firewall*, o *firewall* público é uma ACL de política padrão permissiva e o segundo *firewall*, o *firewall* da sub-rede privada é uma ACL de política padrão restritiva. De acordo com Brodbeck (2016), uma política de permissão libera tudo com exceção do que estiver explicitamente proibido nas regras de bloqueio. Ele também informa que uma política de restrição bloqueia tudo com exceção do que estiver explicitamente permitido nas regras de permissão.

Conforme ilustrado na Figura 29 e Figura 30, a ACL pública foi nomeada como “ACL – Rede Pública” e a ACL privada foi nomeada como “ACL – Rede Privada”, e essas ACL foram associadas a rede VPC “VPC – Rede”.

Figura 29 – Criação ACL pública

Fonte: alterado baseado em AWS (2020)

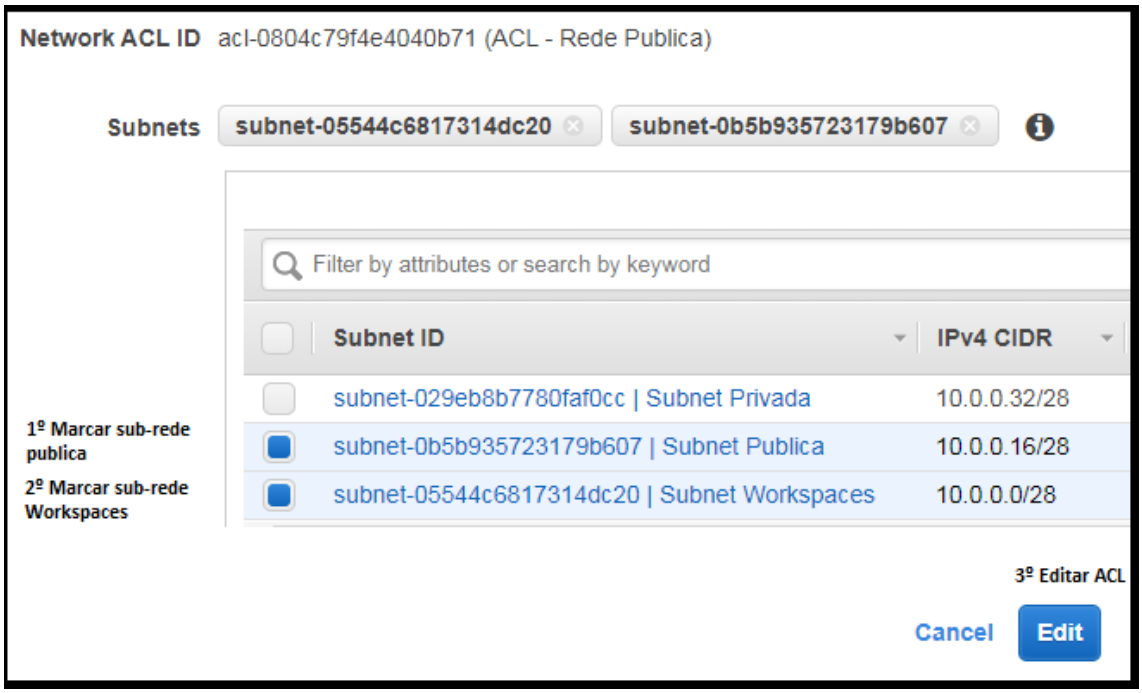
Figura 30 – Criação ACL privada



Fonte: alterado baseado em AWS (2020)

Após a criação das ACL, conforme mostrado na Figura 31, foi associada a sub-rede pública e a sub-rede *Workspaces* a ACL pública, configurando assim que a entrada e saída de dados da sub-rede pública e a sub-rede *Workspaces* devem passar por essa ACL.

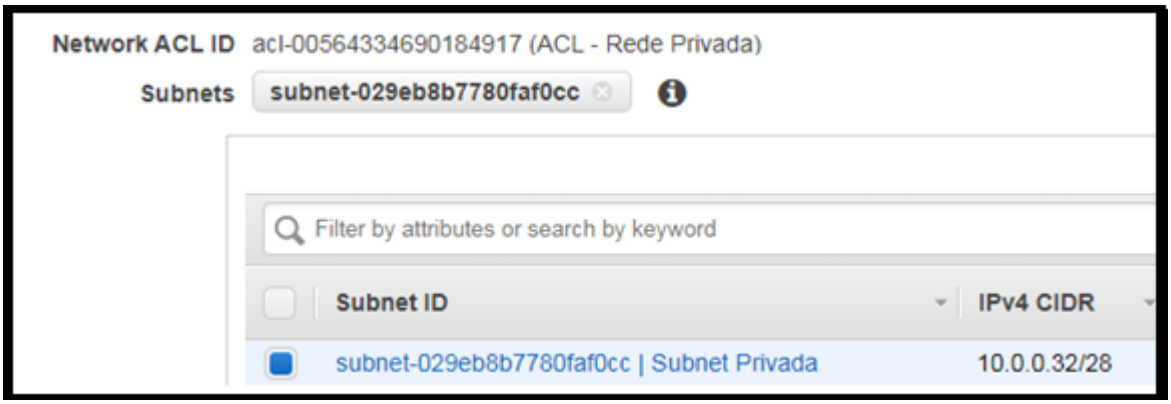
Figura 31 – Associação ACL sub-rede pública e *Workspaces*



Fonte: alterado baseado em AWS (2020)

Conforme mostrado na Figura 32, foi associada a sub-rede privada a ACL privada, configurando assim que a entrada e saída de dados da sub-rede privada deve passar por essa ACL.

Figura 32 – Associação ACL sub-rede privada



Fonte: alterado baseado em AWS (2020)

Após a associação das ACL as sub-redes, foi inserido as regras de entrada e saída, conforme ilustrado na Figura 33. Lembrando que essas regras foram cadastradas como regras de entrada e depois como regras de saída. Elas são cadastradas separadamente.

Figura 33 – Regras de entrada e saída ACL publica

Rule #	Type	Protc	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	157.240.0.0/16	DENY
101	ALL Traffic	ALL	ALL	151.101.178.167/32	DENY
102	ALL Traffic	ALL	ALL	13.227.113.131/32	DENY
103	ALL Traffic	ALL	ALL	157.240.12.0/24	DENY
104	ALL Traffic	ALL	ALL	66.254.114.41/32	DENY
105	ALL Traffic	ALL	ALL	179.191.160.0/19	DENY
106	ALL Traffic	ALL	ALL	104.18.156.3/32	DENY
107	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW

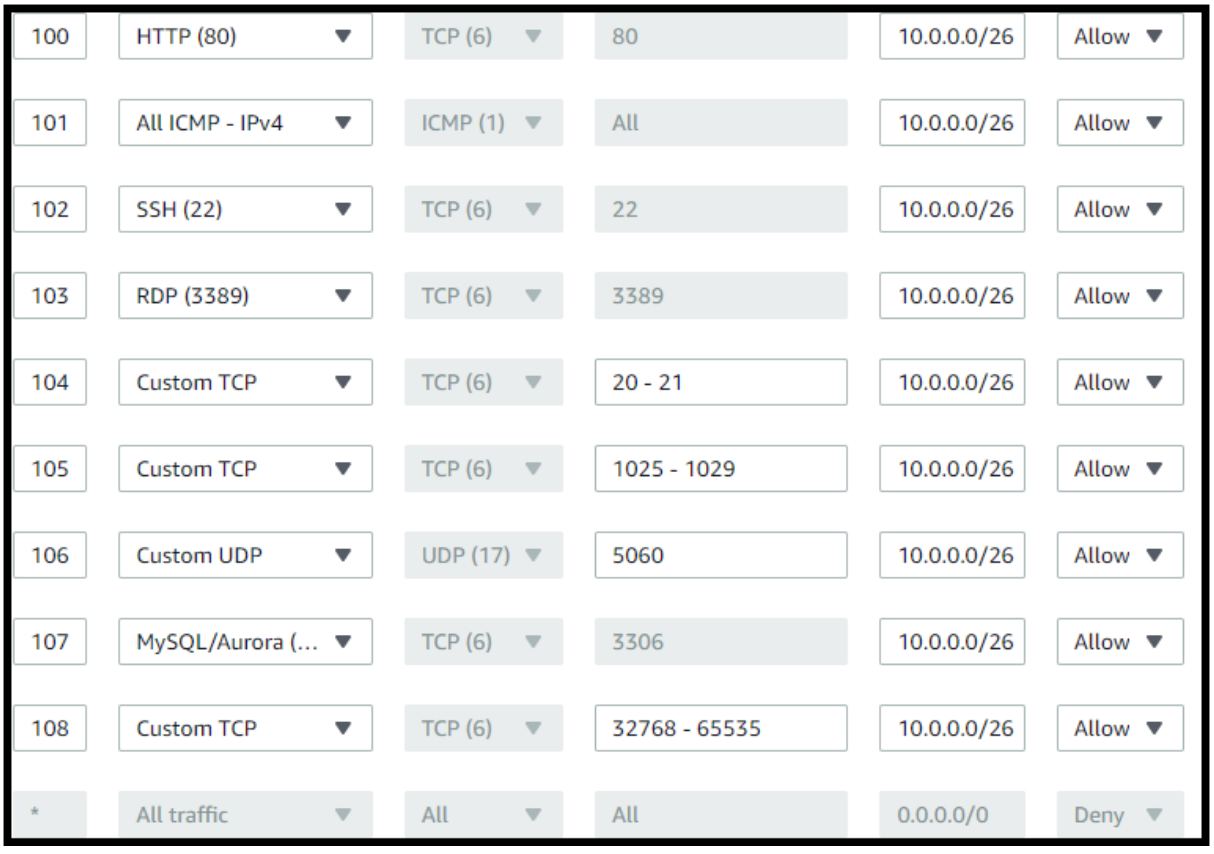
Fonte: alterado baseado em AWS (2020)

As regras apresentadas na Figura 33 são para restringir acesso ao *Facebook, Twitch*, *Amazon Prime Video*, *Instagram*, Baixaki e sites adultos.

Após a inclusão das regras de entrada e saída da ACL publica, foi incluída as regras de entrada e saída da ACL privada. As regras de entrada e saída foram incluídas, conforme mostrado na Figura 34. Lembrando que essas regras foram cadastradas como regras de entrada e depois como regras de saída. Elas são cadastradas separadamente.

Como os clientes VPN, os clientes *Workspaces* e o servidor *Web Proxy* estão conectados à *Internet*, foi criado uma regra permitindo a saída de tráfego por portas efêmeras para permitir que essas instâncias voltadas à *Internet* recebam informações dos servidores. Instâncias *Windows* utilizam as portas 49152-65535 e as instâncias Linux utilizam as portas 32768-61000, portanto foram liberadas as portas 32768-65535 (AWS, 2020).

Figura 34 – Regras de entrada e saída ACL privada



Fonte: alterado baseado em AWS (2020)

As regras de entrada e saída apresentadas na Figura 34, estão permitindo solicitações HTTP, teste de *ping*, conexão SSH, RDP (conexão remota), FTP, DNS e *MySQL* e portas efêmeras, todas elas recebendo tráfego somente da rede VPC.

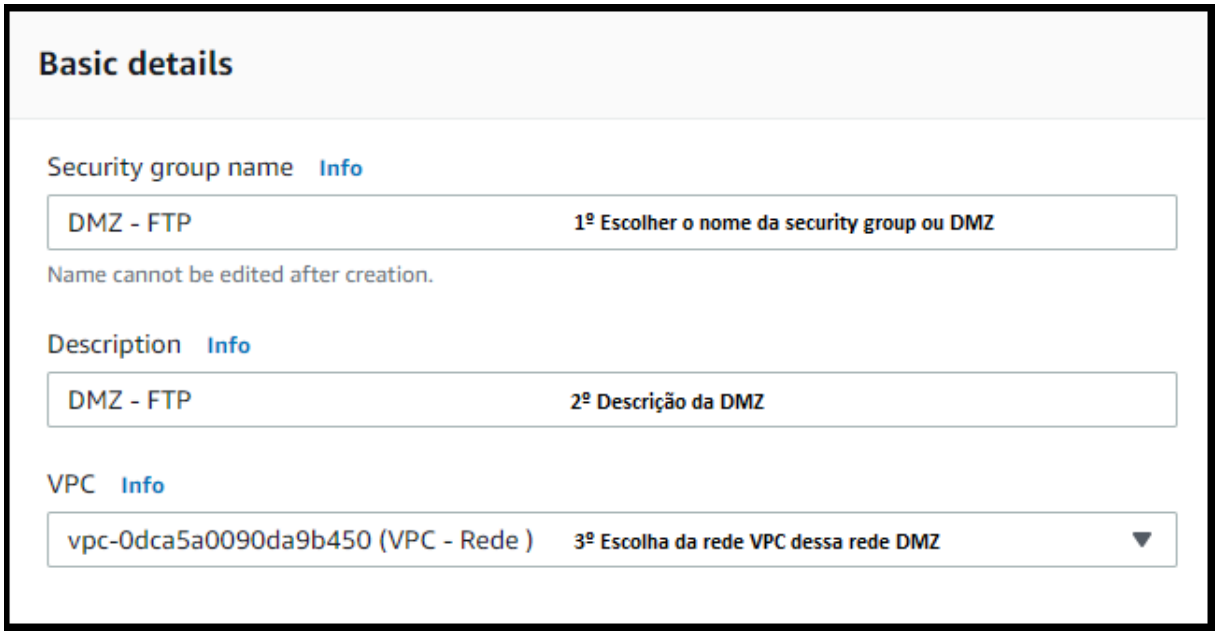
4.2.2 DMZ – *Security Group*

Com exceção dos clientes *Workspaces*, todas as instâncias possuem uma *Security Group* (rede DMZ). Essa decisão foi feita focando mais segurança. Com essa configuração, cada servidor vai receber solicitações pertinentes somente ao que ele desempenha.

4.2.2.1 DMZ – Servidor FTP

A *Security Group* do servidor FTP foi criada conforme apresentado na Figura 35. Essa *Security Group* foi nomeada como “DMZ – FTP”, foi inserido uma descrição falando do que ela é e foi associada a rede “VPC – Rede”.

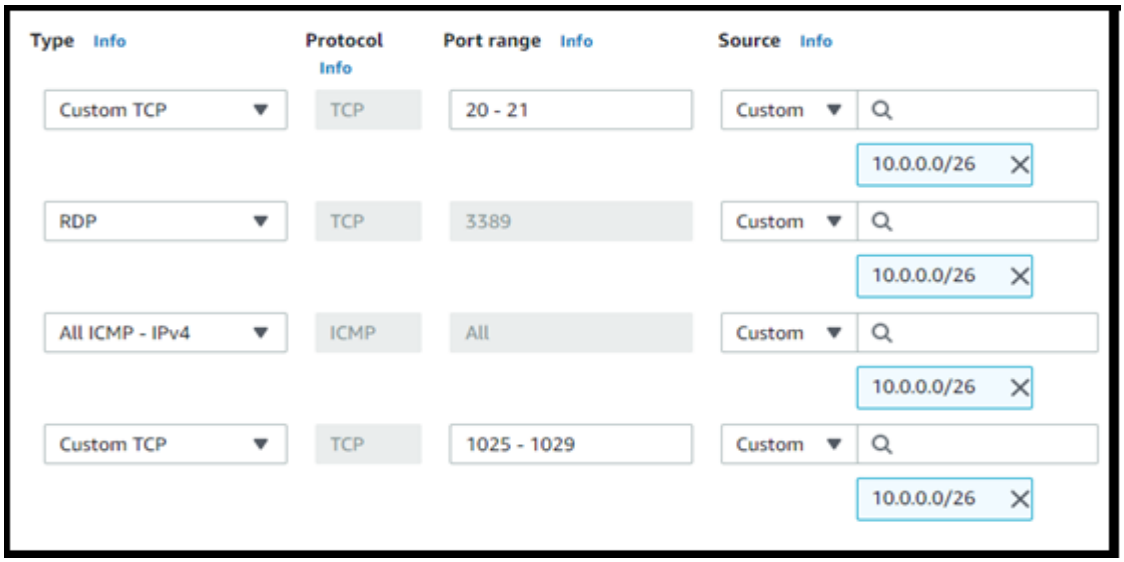
Figura 35 – Criação DMZ servidor FTP



Fonte: alterado baseado em AWS (2020)

Após as configurações informadas na Figura 35, foram inseridas as regras de entrada e saída de tráfego conforme mostrado na Figura 36. As regras de entrada e saída são cadastradas separadamente.

Figura 36 – Regras entrada e saída DMZ FTP



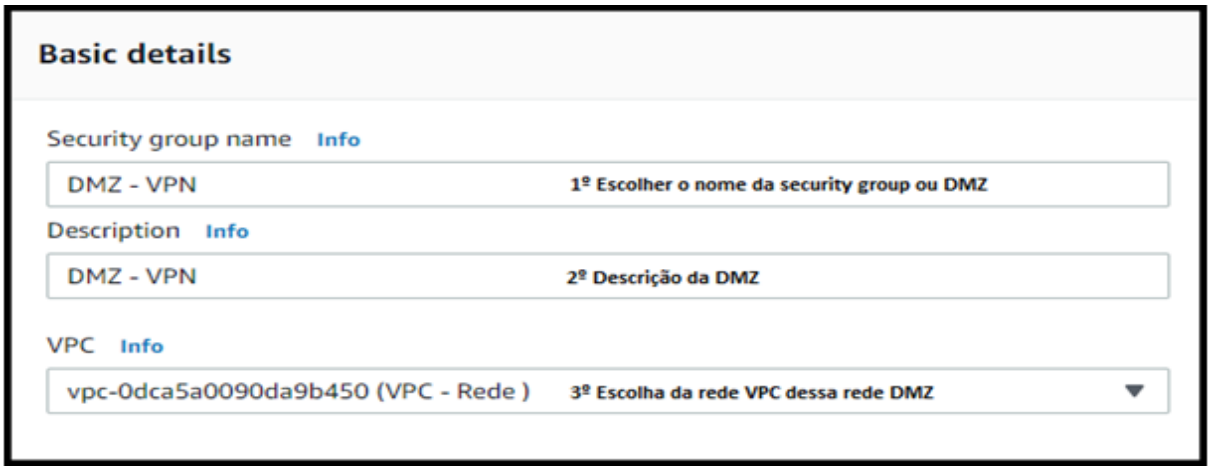
Fonte: alterado baseado em AWS (2020)

Fourozan (2007) informa que o FTP usa porta 20 para transferência de dados e a porta 21 para controle de conexão. Foram escolhidas as portas 1025-1029 para porta de canais de dados. Portanto, a *Security Group* vai permitir conexão FTP, conexão remota e teste de *ping* somente entre a rede VPC.

4.2.2.2 DMZ – Servidor VPN

A *Security Group* do servidor VPN foi criada conforme mostrado na Figura 37. Essa *Security Group* foi nomeada como “DMZ – VPN”, foi inserido uma descrição falando do que ela é e foi associada à rede VPC “VPC-Rede”.

Figura 37 – Criação DMZ servidor VPN



Fonte: alterado baseado em AWS (2020)

Após as configurações informadas na Figura 37, foram inseridas as regras de entrada conforme ilustrado na Figura 38.

Figura 38 – Regras de entrada DMZ VPN

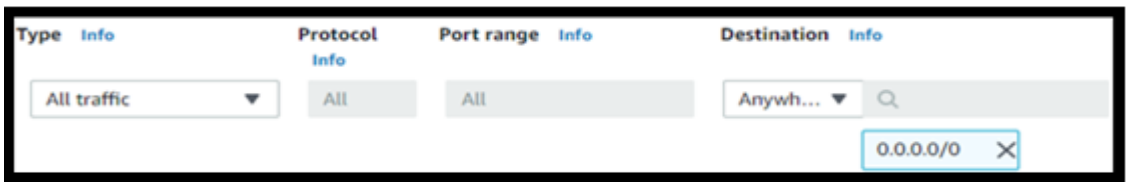
Fonte: alterado baseado em AWS (2020)

Conforme passo a passo para configuração de servidor *OpenVPN* na AWS, foi habilitado passagem de tráfego pelas portas:

- SSH;
- HTTPS;
- TCP 943;
- UDP 1194;

Essas são portas necessárias para conexão VPN. Além delas ainda incluímos porta para testes de *ping* por fins de teste (AWS, 2020). Já nas regras de saída, como mostrado na Figura 39, foi permitida a saída de todo tráfego indo para qualquer IP.

Figura 39 – Regras de saída DMZ VPN

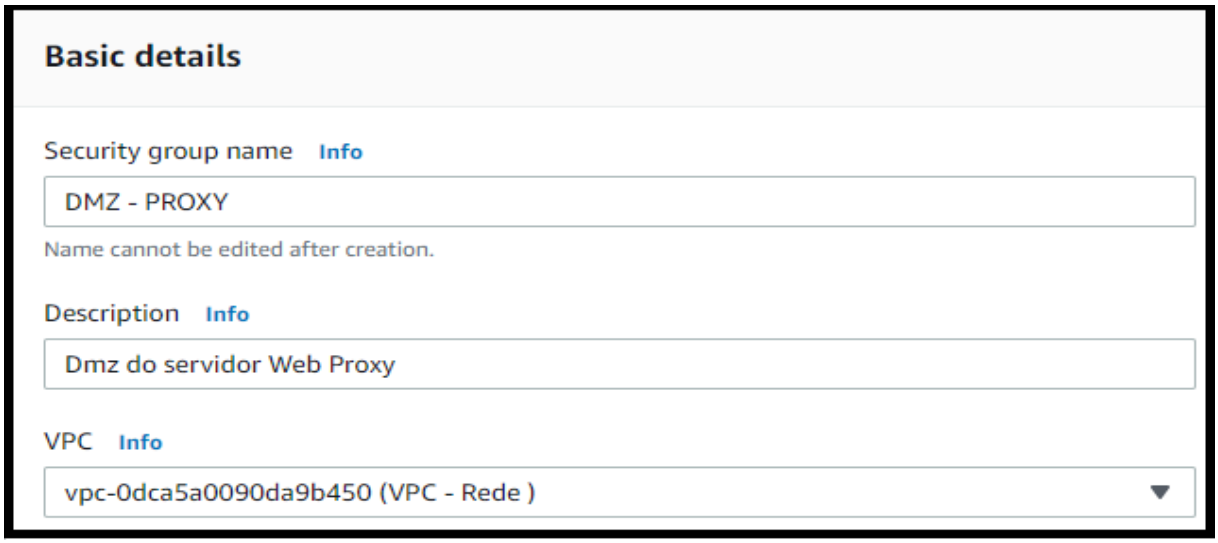


Fonte: alterado baseado em AWS (2020)

4.2.2.3 DMZ – Servidor *Web Proxy*

A *Security Group* do servidor *Web Proxy* foi criada conforme mostrado na Figura 40. Essa *Security Group* foi nomeada como “DMZ – PROXY”, na qual foi inserida uma descrição, informando sobre ela e foi associada à rede VPC “VPC-Rede”.

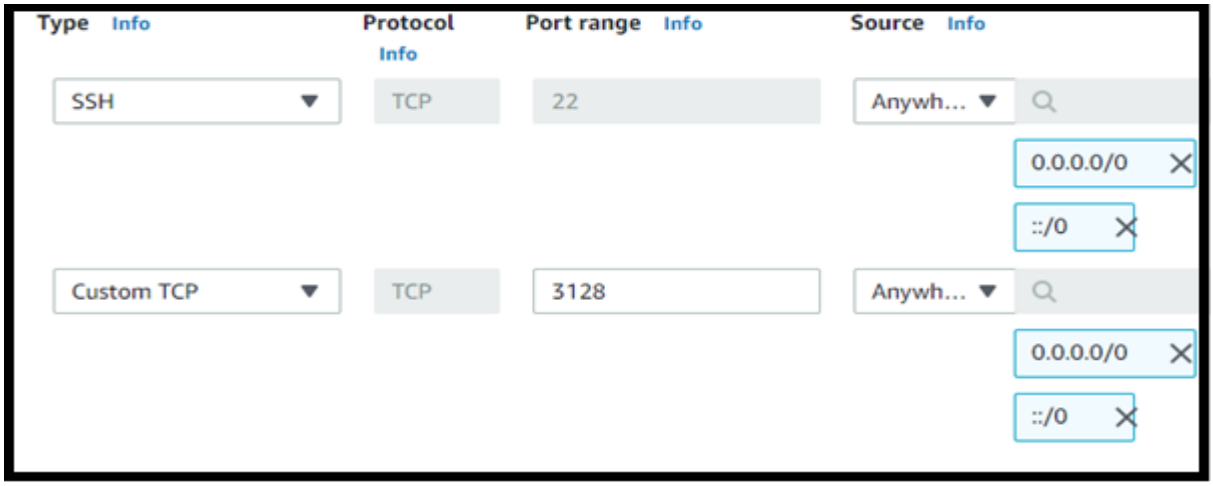
Figura 40 – Criação DMZ servidor *Web Proxy*



Fonte: alterado baseado em AWS (2020)

Foram incluídas regras de entrada, conforme mostrado na Figura 41. Foi liberada passagem de dados pela porta SSH e pela porta 3128, vindo de qualquer IP. A porta 3128 foi escolhida como porta de conexão com o servidor *Web Proxy*.

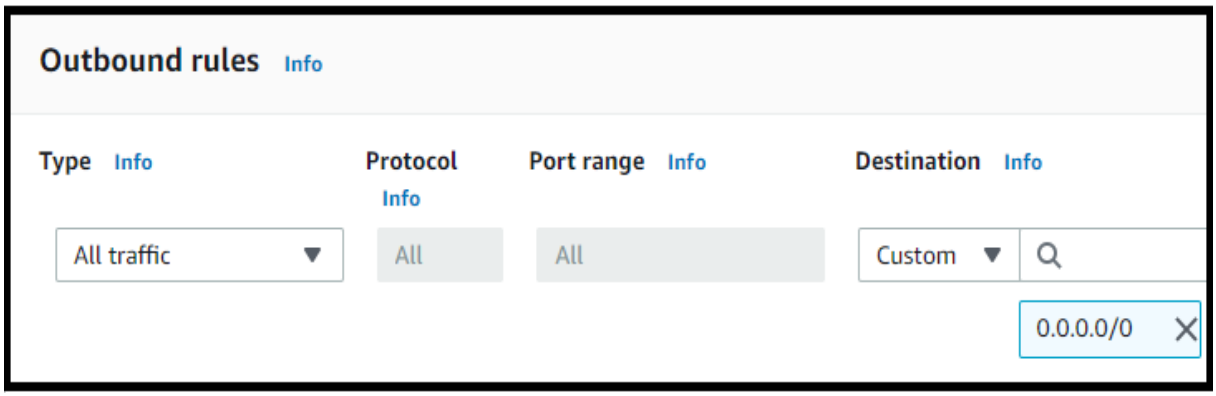
Figura 41 – Regras de entrada DMZ *Web Proxy*



Fonte: alterado baseado em AWS (2020)

Para as regras de saídas, foi liberado tráfego saindo para qualquer IP e qualquer porta, como apresentado na Figura 42.

Figura 42 – Regras de saída DMZ *Web Proxy*

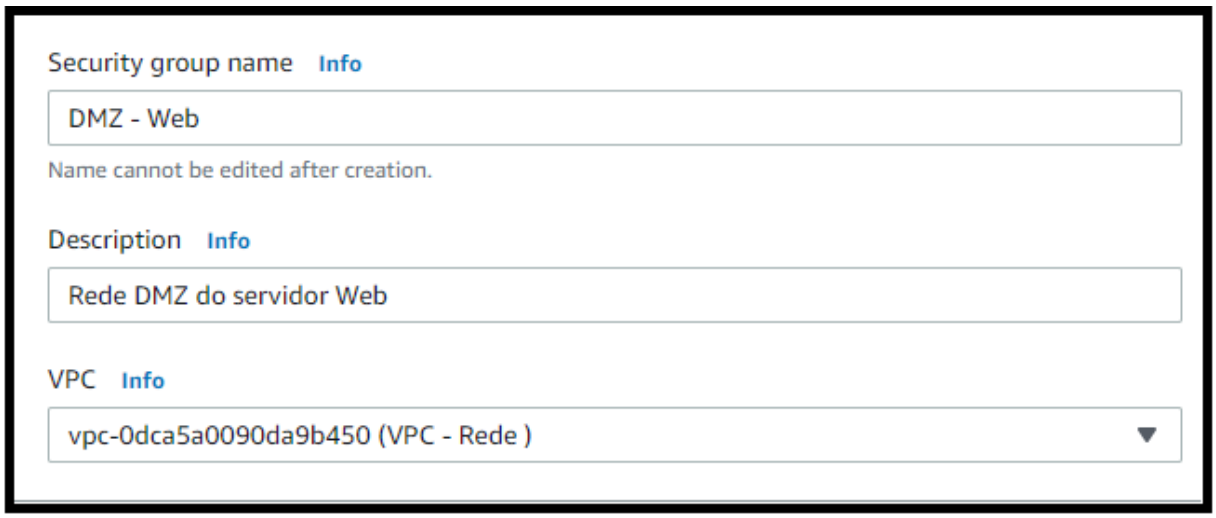


Fonte: alterado baseado em AWS (2020)

4.2.2.4 DMZ – Servidor *Web*

A *Security Group* do servidor *Web* foi criada conforme ilustrado na Figura 43, sendo nomeada como “DMZ – Web”. Foi inserida uma descrição informando sobre ela e foi associada à rede “VPC-Rede”.

Figura 43 – Criação DMZ servidor *Web*

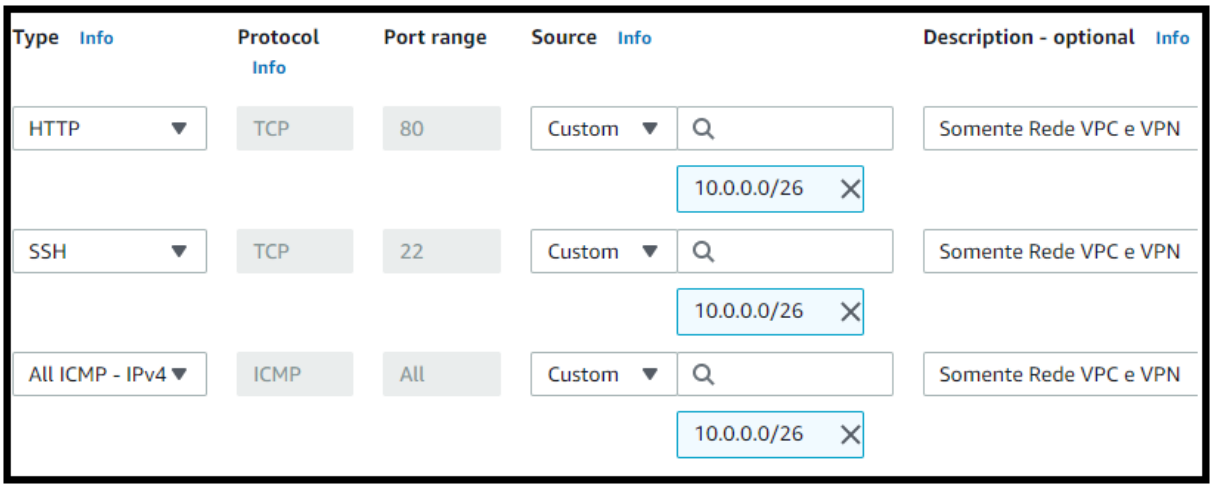


Fonte: alterado baseado em AWS (2021)

Como mostrado na Figura 44, foi liberado tráfego de entrada nas portas de conexão HTTP, SSH e teste de *ping*, pois esse servidor necessita receber dados

somente via HTTP com as demais instâncias. O tráfego é permitido somente vindo da rede VPC.

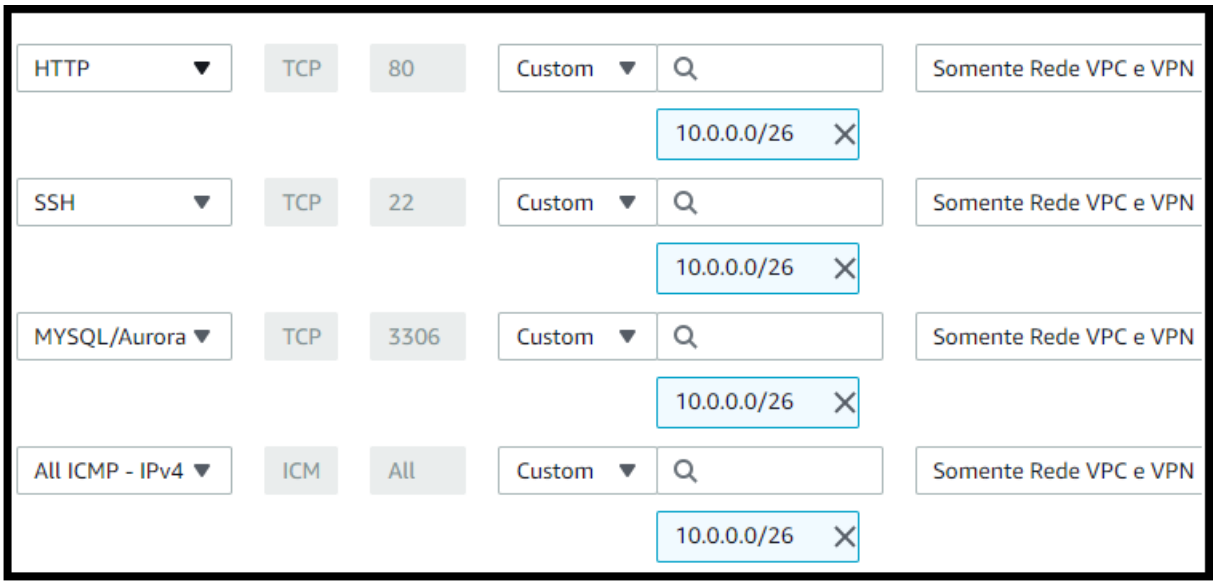
Figura 44 – Regras de entrada DMZ *Web*



Fonte: alterado baseado em AWS (2021)

Conforme apresentado na Figura 45, foi liberado tráfego de saída nas portas de conexão HTTP, SSH, *MySQL* e teste de *ping*, pois esse servidor se comunica via HTTP com as demais instâncias e via porta TCP 3306 com o banco de dados *MySQL*. O tráfego é permitido somente saindo para a rede VPC.

Figura 45 – Regras de saída DMZ *Web*

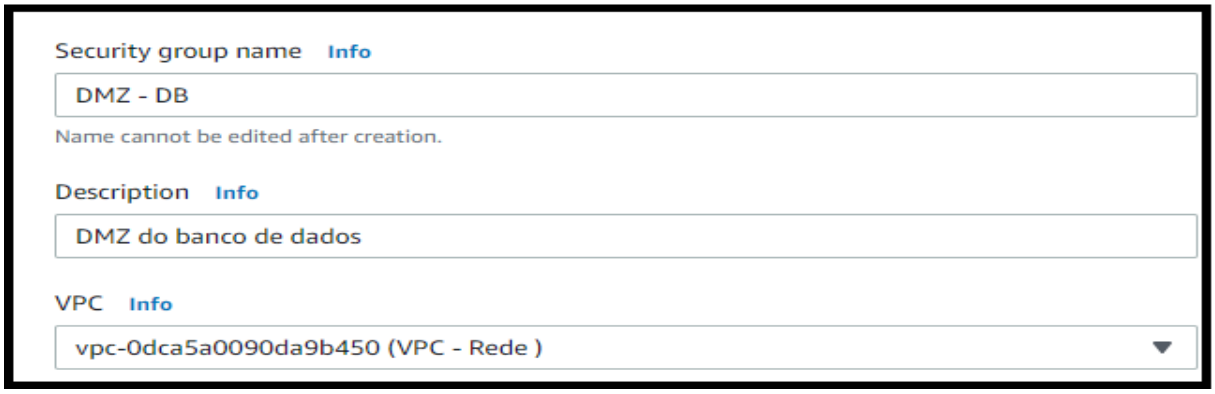


Fonte: alterado baseado em AWS (2021)

4.2.2.5 DMZ – Banco de dados RDS

A *Security Group* do banco de dados foi criada conforme ilustrado na Figura 46, sendo nomeada de “DMZ – DB”. Foi inserida uma descrição informando sobre ela e foi associada à rede “VPC-Rede”.

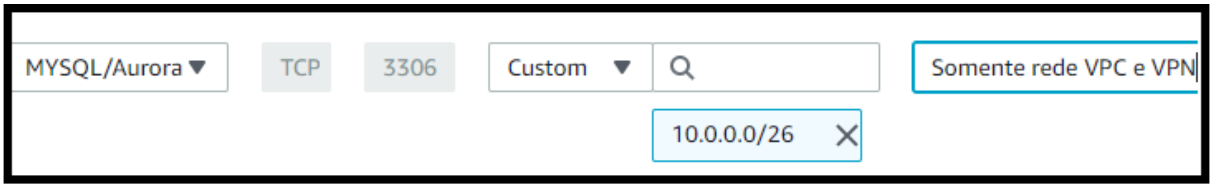
Figura 46 – Criação DMZ do banco de dados



Fonte: alterado baseado em AWS (2021)

Como mostrado na Figura 47, essa DMZ permite tráfego de entrada somente na porta 3306 - que é a porta *default* do *MySQL -,* vindo da rede VPC.

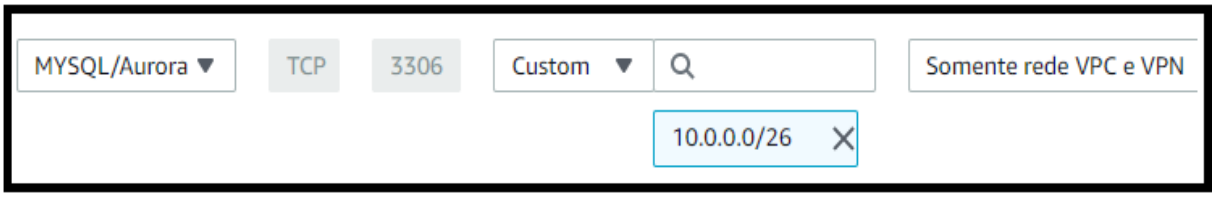
Figura 47 – Regras de entrada DMZ banco de dados



Fonte: alterado baseado em AWS (2021)

Como apresentado na Figura 48, essa DMZ permite tráfego de saída somente na porta 3306, vindo da rede VPC.

Figura 48 – Regras de saída DMZ banco de dados



Fonte: alterado baseado em AWS (2021)

4.3 Servidores e *Desktops*

Neste trabalho foram implementados quatro servidores: o servidor VPN, servidor FTP, servidor *Web* e o servidor *Web Proxy* e dois clientes *Workspaces,* conforme mostrado na Figura 1 da página 40.

4.3.1 Servidor *Web Proxy;*

O servidor *Web Proxy* foi criado para restringir o acesso aos sites do *Twitter*, *Netflix* e *Disney* dos clientes VPN e *Workspaces* em nível de interface e por mascarar o IP desses clientes.

Para criação da instância do servidor *Web Proxy,* foram utilizados os passos mostrados no anexo D – Utilizando a AMI Ubuntu 18.04 e escolhendo a DMZ "DMZ – PROXY" - e foi utilizado o par de chaves, gerado na criação do servidor VPN. A instância foi acessada conforme mostrado no Anexo B.

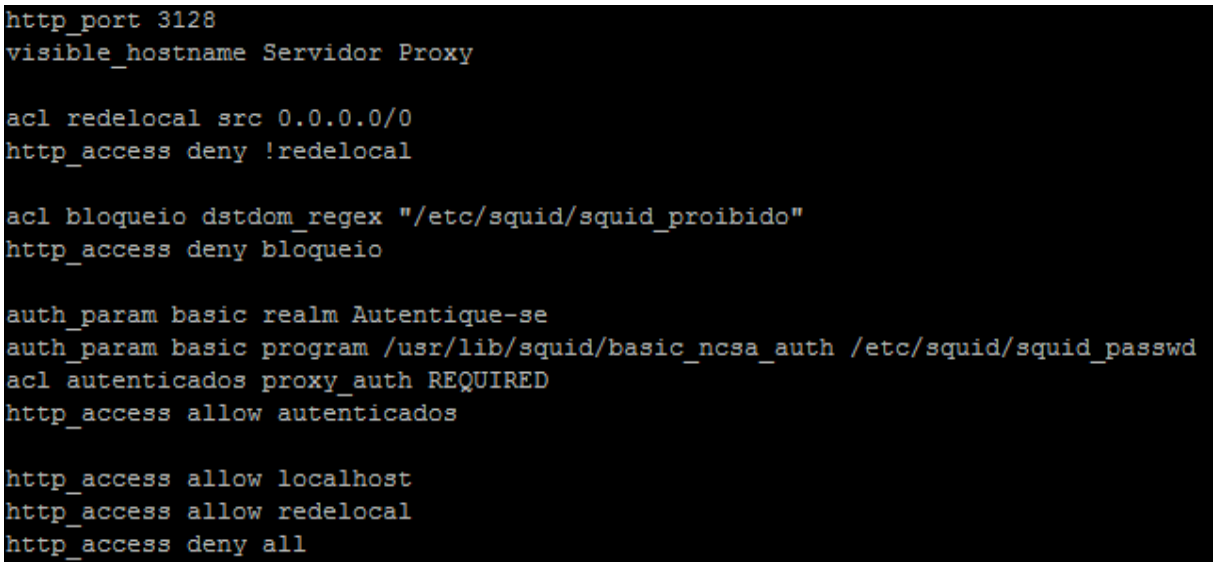
Foi realizado *login* com permissões de administrador utilizando o comando "*sudo su".* Foi realizado *download* do aplicativo *Squid* utilizando o comando "*apt install squid*". Também foi realizado o *download* de utilitários do *apache2* usando o comando "*apt install apache2-utils*" para usar o comando "*htpasswd*" que utiliza encriptação de senha utilizando o algoritmo MD5 (APACHE, 2020).

Na pasta "/etc/squid" foram criados três arquivos, são eles:

- squid_passwd: Arquivo que contém as informações dos usuários do servidor *Web Proxy*, tais como seu *login* e sua senha. A senha está encriptada utilizando o algoritmo MD5
- squid_proibido: Arquivo que contém as palavras que proibirão o acesso a certos sites.
- squid.conf: Arquivos que contêm as configurações do servidor *Web Proxy*.

Para criação de novos usuários, foi utilizado o comando "*htpasswd /etc/squid/squid_passwd* 'nome do novo usuário'". Foi criado usuário com o *login* "yann.sml" e "gabryel.sml". As palavras proibidas adicionadas no arquivo squid_proibido foram *Netflix, twitter* e *disney*. As instruções utilizadas no arquivo squid.conf estão apresentadas na Figura 49.

Figura 49 – Instruções squid.conf



Fonte: alterado baseado em Ubuntu (2020)

Na Figura 49 é apresentado:

- Na primeira linha é informada a porta 3128 para comunicação.
- Na segunda linha é informado qual será o nome deste servidor, foi informado como “Servidor *Proxy*”.
- A terceira linha informa que esse servidor pertence à *Internet*.
- A quarta linha informa que toda a *Internet* consegue acesso ao servidor *Web Proxy*.
- Na quinta linha é informado que ao abrir o navegador, deve aparecer a mensagem “Autentique-se”.
- Na sexta linha é informado que é para ser utilizada autenticação por *login* e senha, e qual o arquivo que possui os usuários.
- A sétima linha informa a lista de usuários “autenticados” que foram autenticados corretamente.
- Na oitava linha é informado que é permitido o acesso de todos os usuários que estão na lista de usuários “autenticados”.
- A nona, decima e decima primeira linha informa que se for autenticado corretamente, esse servidor e toda a *Internet* consegue *logar* no servidor *Web Proxy*.

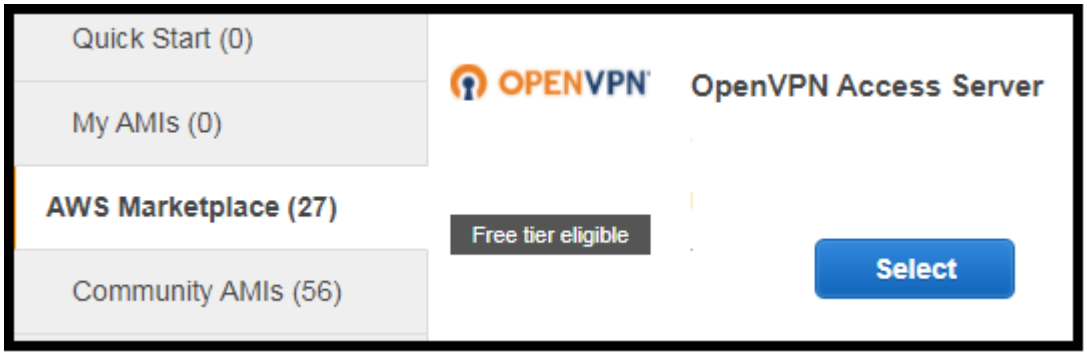
Por fim, foi utilizado o comando “*service squid restart*” para ligar o servidor.

4.3.2 Servidor VPN

O servidor VPN foi criado para que clientes na *Internet* possam ter acesso aos servidores da sub-rede privada. Foi criado o servidor VPN e dois clientes VPN.

Foi escolhido o serviço “EC2” da AWS e escolhido a opção “Executar Instância”. Conforme mostrado na Figura 50, foi escolhida a AMI “*OpenVPN* Access Server” pois é um SO feito com propósito de criação de servidores VPN.

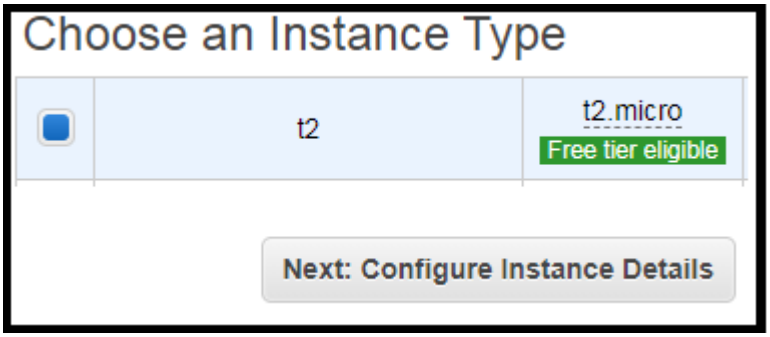
Figura 50 – Seleção AMI VPN



Fonte: alterado baseado em AWS (2020)

Na Figura 51 é mostrado a seleção da instância VPN como CPU do tipo “t2.*micro*” devido a sua gratuidade.

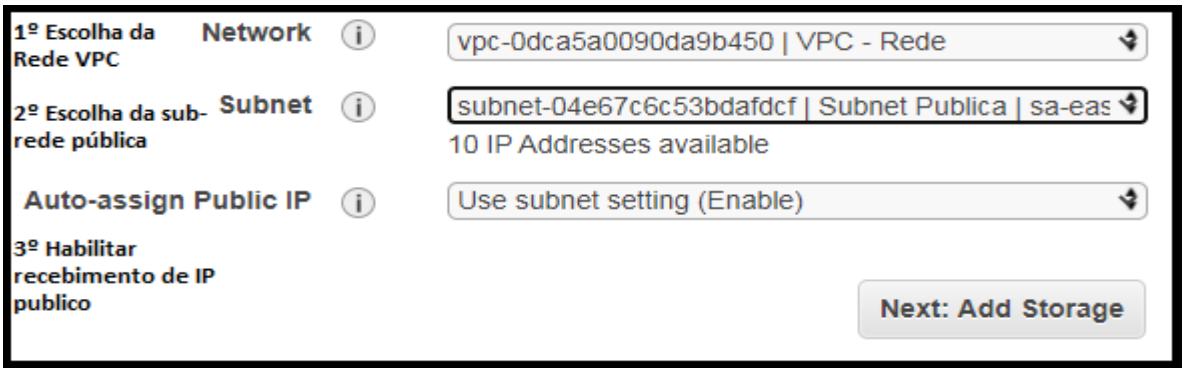
Figura 51 – Seleção configuração de instância VPN



Fonte: alterado baseado em AWS (2020)

. A Figura 52 apresenta a seleção da rede VPC “VPC-Rede”, a seleção da sub-rede Pública e a habilitação da atribuição automática de IP público.

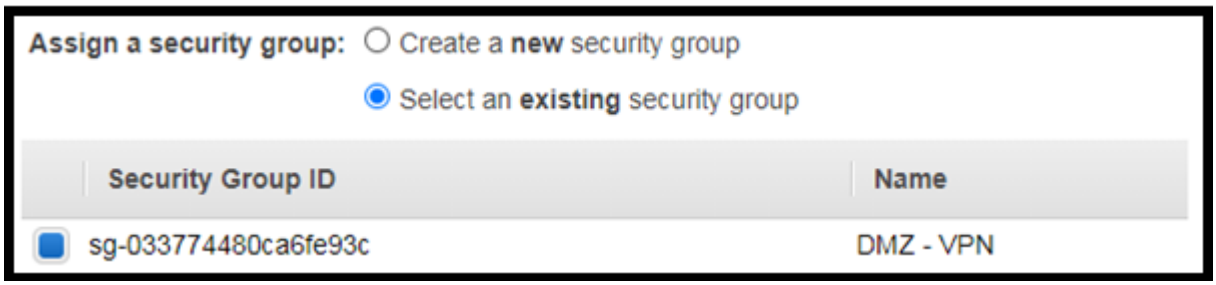
Figura 52 – Seleção rede e sub-rede VPN



Fonte: alterado baseado em AWS (2020)

Então, conforme apresentado na Figura 53, foi associada a essa instância a *Security Group* “DMZ-VPN”.

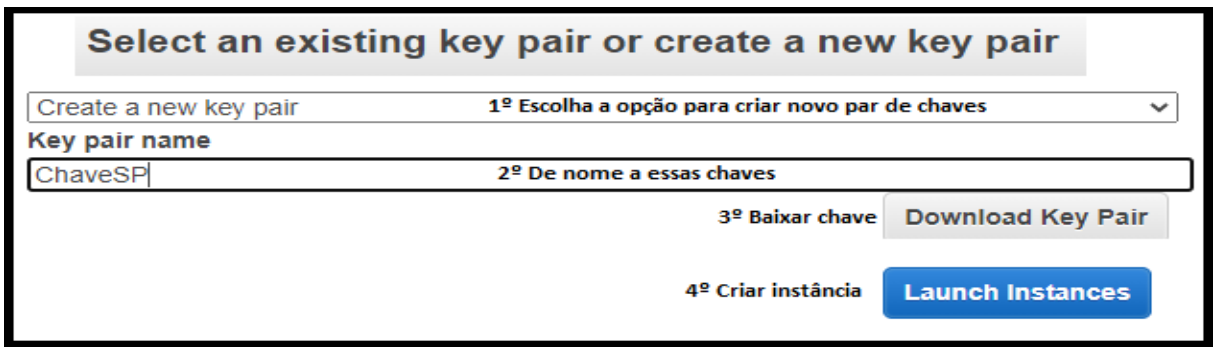
Figura 53 – Seleção *Security Group* VPN



Fonte: alterado baseado em AWS (2020)

Por fim, conforme apresentado na Figura 54, foi criado um par de chaves – que foi usada para acesso SSH no servidor – e a confirmação da criação da instância.

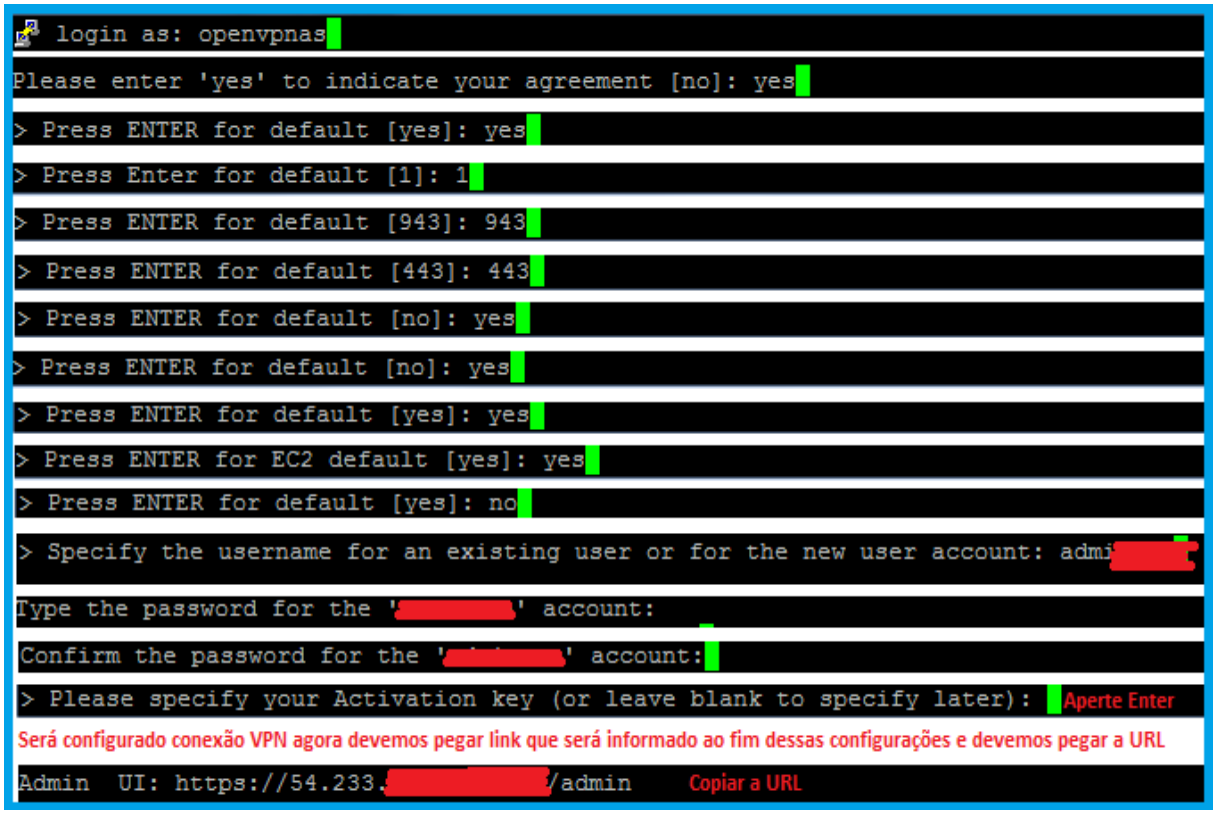
Figura 54 – Criação do par de chaves



Fonte: alterado baseado em AWS (2020)

A instância foi acessada conforme mostrado no Anexo B. Foi informado os comandos usados na Figura 55 para configuração do servidor VPN.

Figura 55 – Comandos para configuração do servidor VPN



Fonte: alterado baseado em OpenVPN (2020)

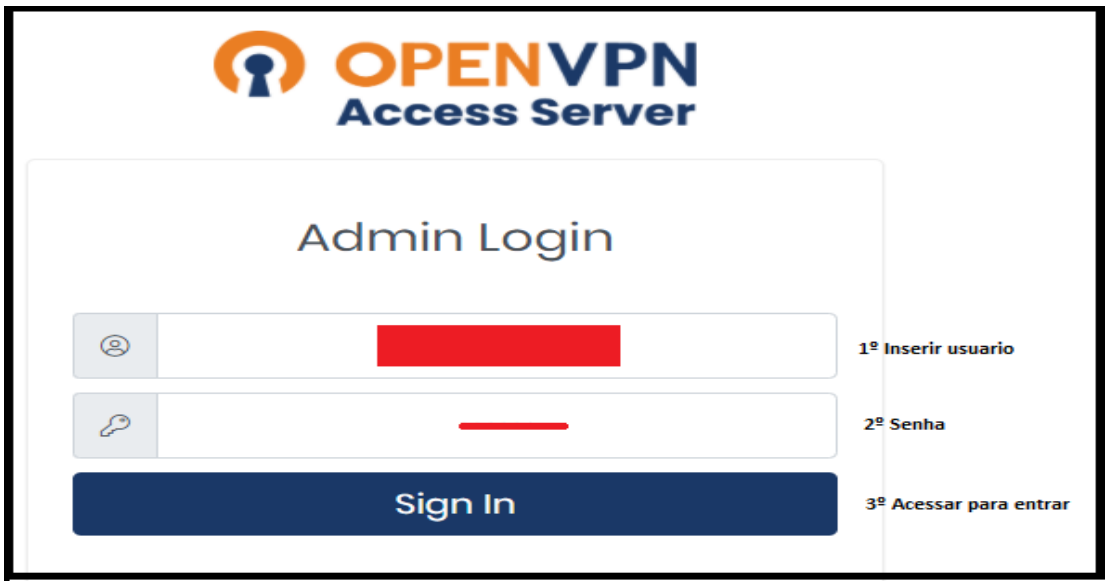
Na Figura 55 é mostrado que:

- Na primeira linha foi informada o usuário padrão do *OpenVPN*.
- Na segunda linha é informado que se está de acordo com as normas do *OpenVPN*.
- Na terceira linha é informado que este foi o nó de acesso primário.
- Na quarta linha é informado que a VPN deve utilizar todas as interfaces e utilizar o IP privado recebido pela plataforma AWS.
- Na quinta linha foi especificada a porta 943 como a porta de uso do *Admin Web UI*.
- Na sexta linha foi especificado a porta 443 como a porta para uso do *OpenVPN*.
- Na sétima linha foi especificado que o tráfego do cliente deve ser roteado por padrão através da VPN.

- Na oitava linha foi informado que o tráfego DNS do cliente deve ser roteado por padrão, através da VPN.
- Na nona linha foi informado que é necessária a autenticação local para acesso ao banco de dados interno.
- Na decima linha é apresentado qual rede interna ele pertence – rede VPC cujos endereços IP são 10.0.0.0/26 – e é informado que as sub-redes privadas dessa rede pode ser acessada pelos clientes VPN.
- Na decima primeira linha é perguntado se é desejado continuar com usuário administrador padrão e é informado que não.
- Na decima segunda linha é informado o novo *login*.
- Na decima terceira linha a nova senha.
- Na decima quarta linha é confirmada essa senha.
- Na decima quinta linha é perguntado se deseja cadastrar chave de ativação agora e é informado *enter* para cadastrar chave de ativação mais tarde, pois essa chave não foi obtiva.
- A última linha mostra a URL do servidor VPN e essa URL foi copiada para o próximo passo.

Essa URL foi colada no *Google Chrome* e é acessada a página ilustrada na Figura 56. Nessa página foi informado o *login* e a senha, criadas a partir dos comandos da etapa anterior

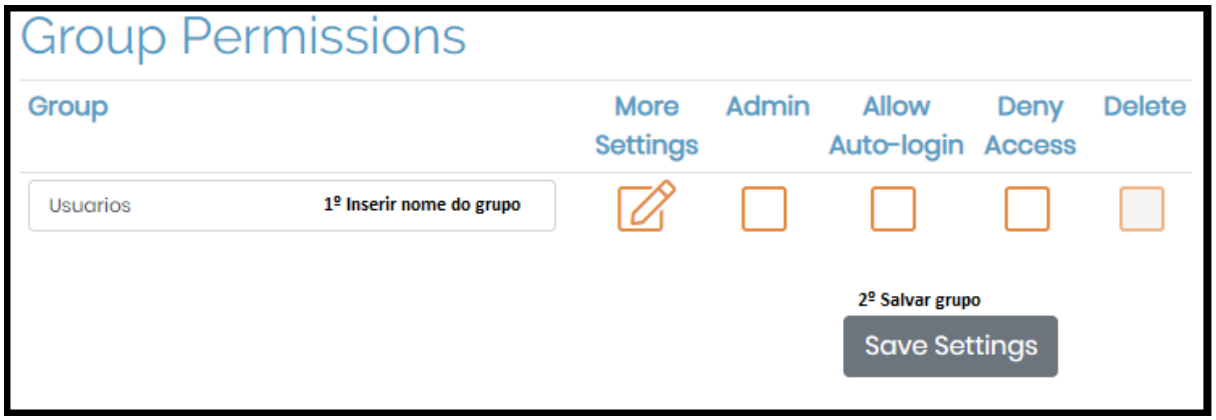
Figura 56 – *Login* no site de configuração VPN



Fonte: alterado baseado em OpenVPN (2020)

Foi acessada aba “*Group Permissions*” para criar um grupo de usuários. É informado o nome desse novo grupo chamado “Usuarios” e, então, este grupo foi criado conforme ilustrado na Figura 57.

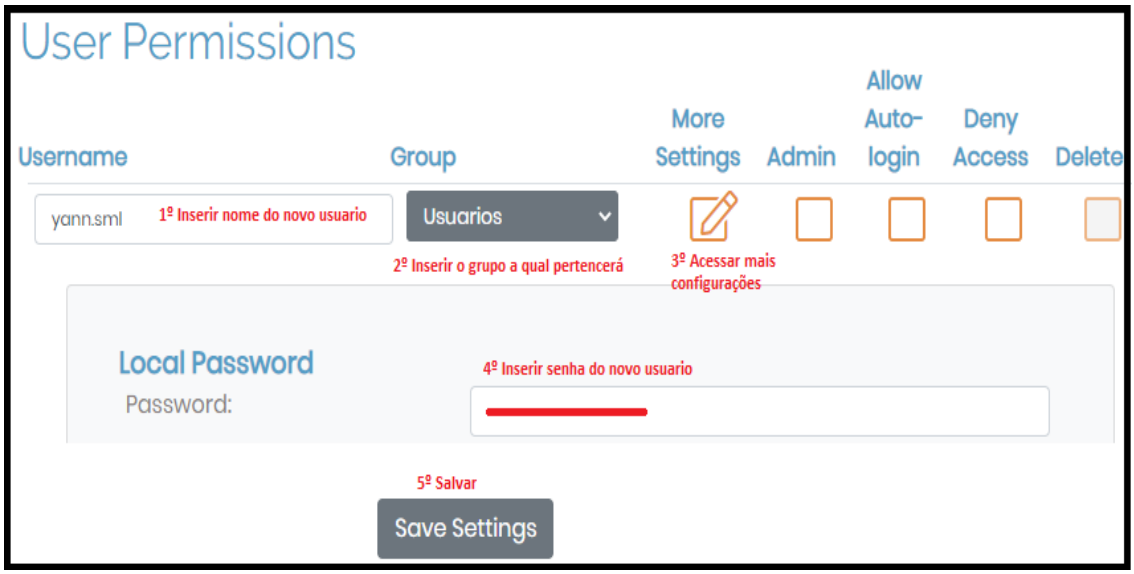
Figura 57 – Criação novo grupo VPN



Fonte: alterado baseado em OpenVPN (2020)

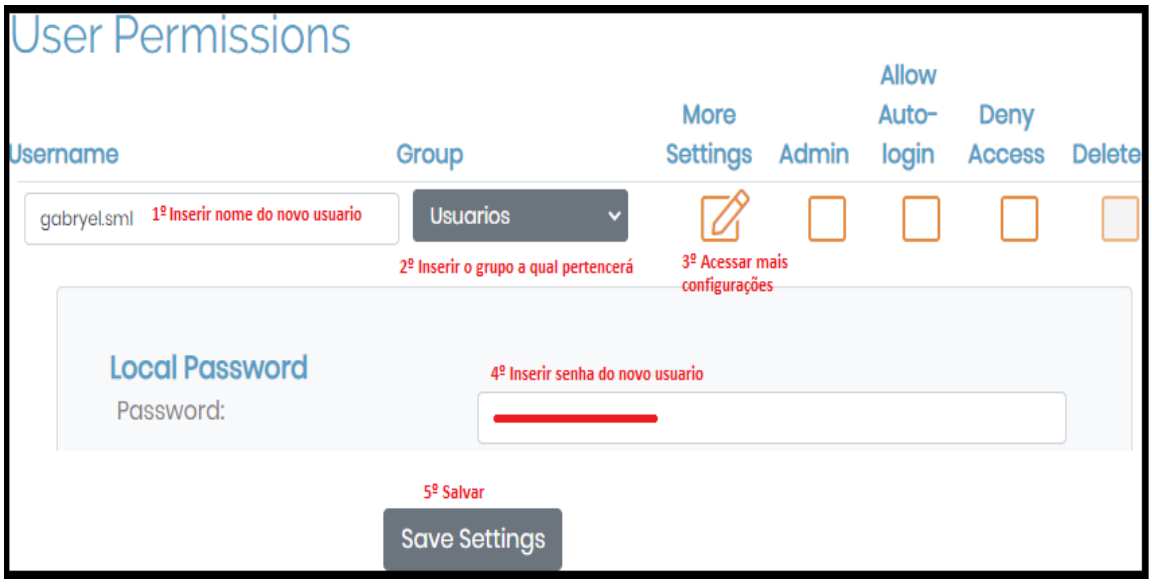
Foi acessada a aba “*User Permissions*” para criar usuários. Conforme ilustrado nas Figuras 58 e 59, foram criados dois usuários para pertencentes ao Grupo “Usuários”.

Figura 58 – Criação usuário yannsml VPN



Fonte: alterado baseado em OpenVPN (2020)

Figura 59 – Criação usuário gabryelsml VPN



Fonte: alterado baseado em OpenVPN (2020)

4.3.3 Servidor FTP

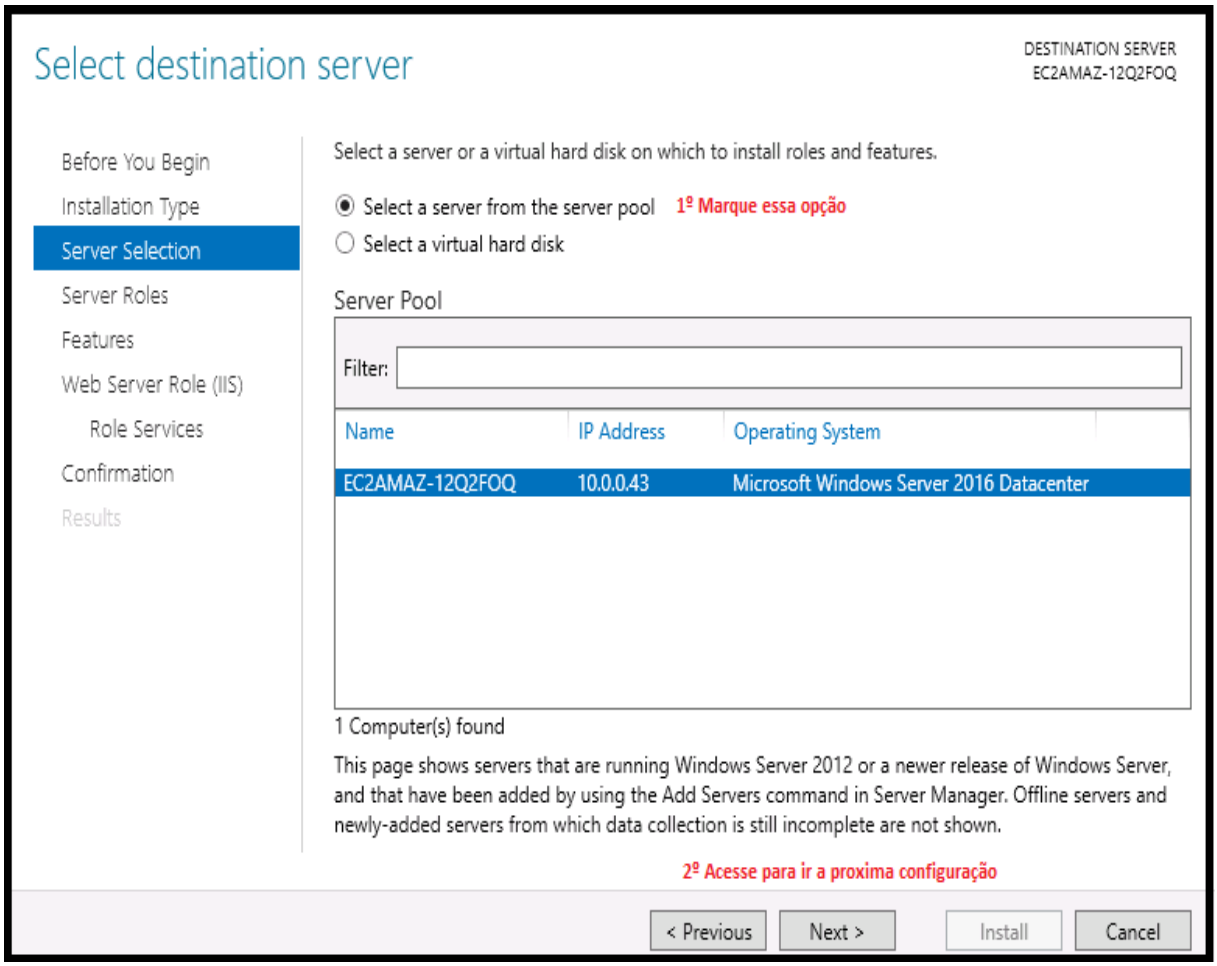
Esse servidor FTP foi criado para armazenar e transferir arquivos dos clientes.

Para criação da instância do servidor FTP foram utilizados os passos mostrados no anexo D, utilizando a AMI *Windows Server 2016,* escolhendo a DMZ “DMZ – FTP” e foi utilizado o par de chaves criado na criação do servidor VPN.

Para acesso remoto do servidor FTP, primeiramente, foi realizado conexão VPN para possuir acesso aos servidores da sub-rede privada. Depois foram utilizados os passos mostrados no Anexo C. Como é necessário baixar os recursos de servidor *Web*, foram utilizados os passos do Anexo A, fazendo assim o servidor FTP possuir acesso à *Internet*.

A seguir, foi acessado o “Gerenciador de Servidor” do *Windows Server* e selecionada a aba “Adicionar funções e recursos”. Foi marcada a opção de instalação, baseada em funções ou recursos. Foi selecionada a própria máquina para instalar os recursos, conforme ilustrado na Figura 60.

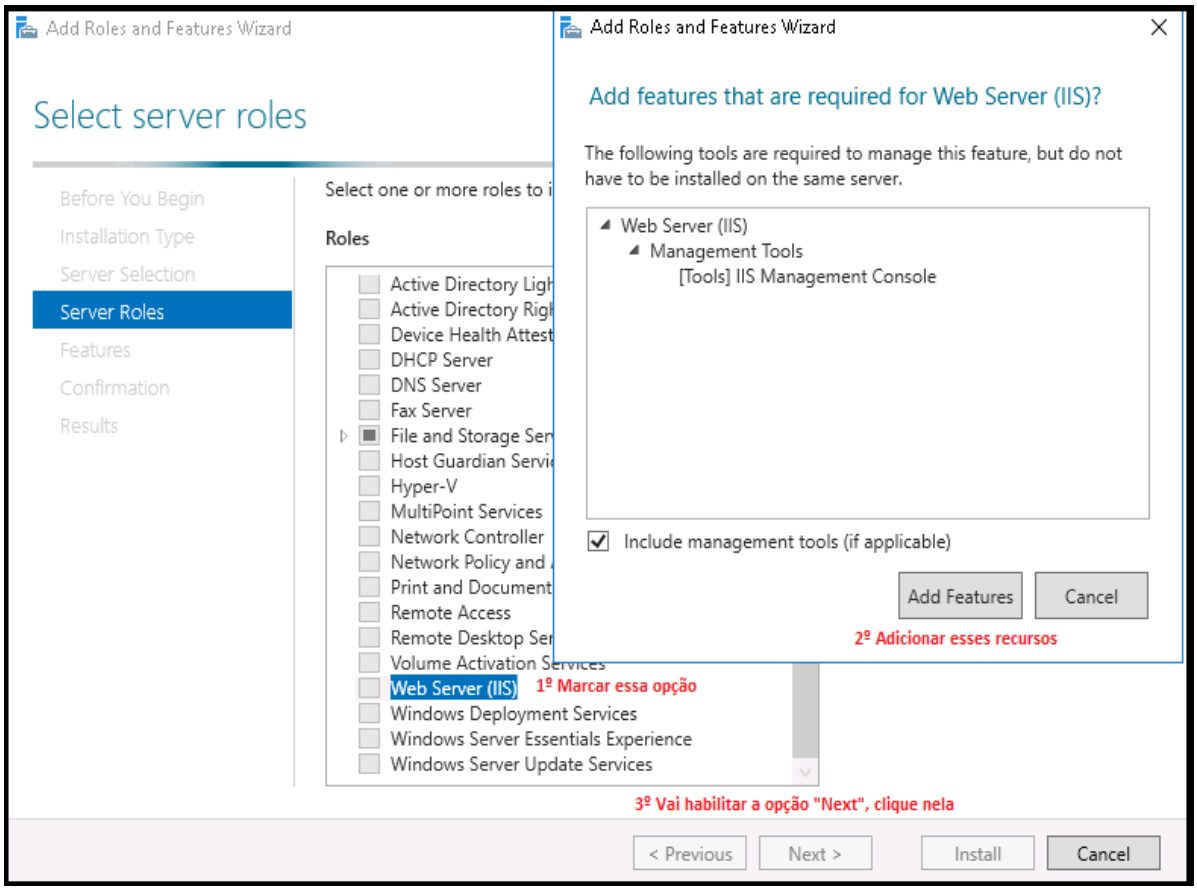
Figura 60 – Seleção para criação servidor FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Foi marcada a opção “Servidor Web” para instalação de recursos sobre essa função, conforme mostrado na Figura 61.

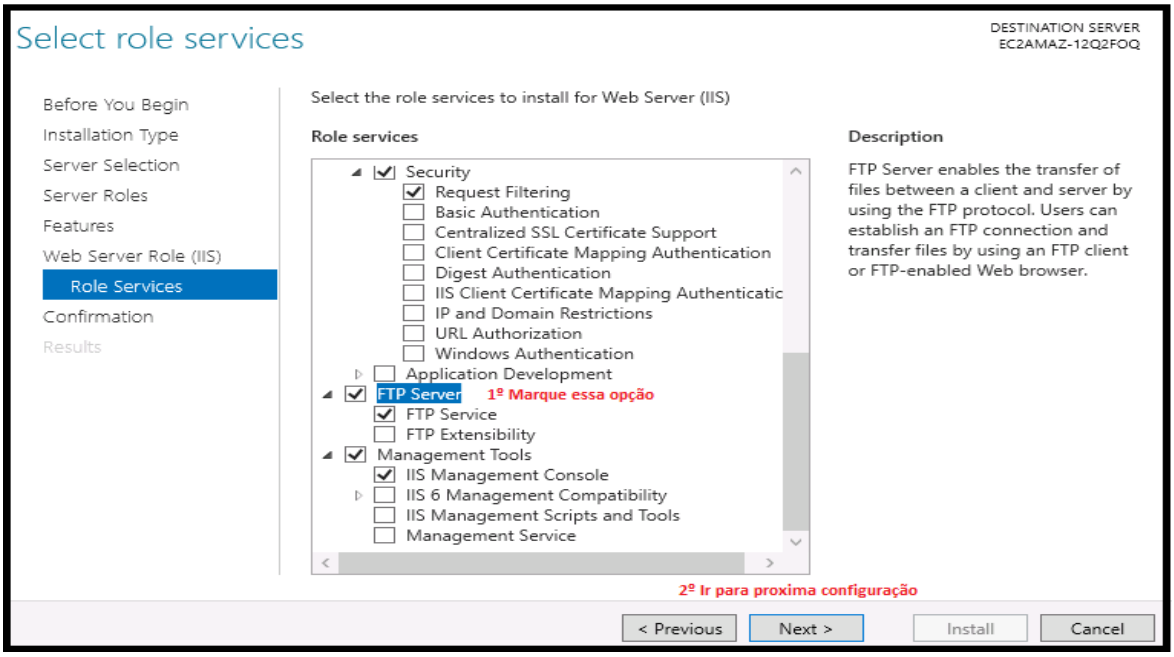
Figura 61 – Marcação servidor *Web Server* FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Em funções de serviço, foi selecionada a opção “Servidor FTP”, conforme ilustrado na Figura 62 e então foram instalados os recursos que serão utilizados para a criação do servidor FTP. Após essas configurações foi retirado o acesso à *Internet*, utilizando passos do Anexo A.

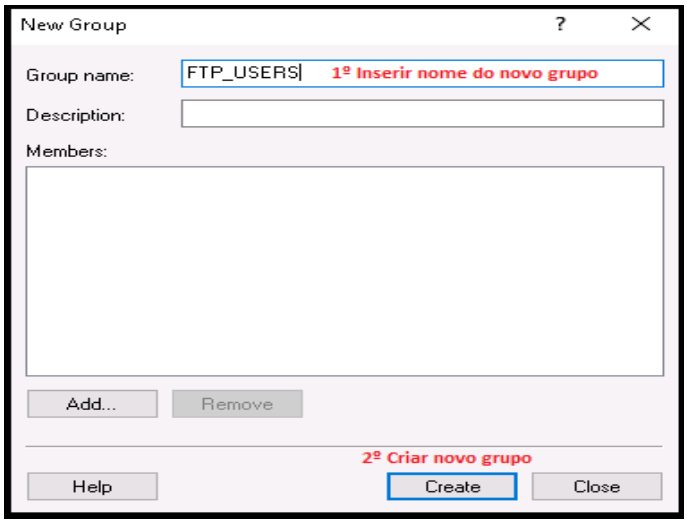
Figura 62 - Marcação Servidor FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Foi acessado o aplicativo “Gerenciamento de computador” do *Windows Server 2016.* Foi acessado a opção “Usuários e Grupos local”, com o botão direito foi escolhida a aba “Grupos” e selecionou-se a opção “Novo Grupo”. Conforme ilustrado na Figura 63, o novo grupo foi nomeado como “FTP_USERS” e, então, criado.

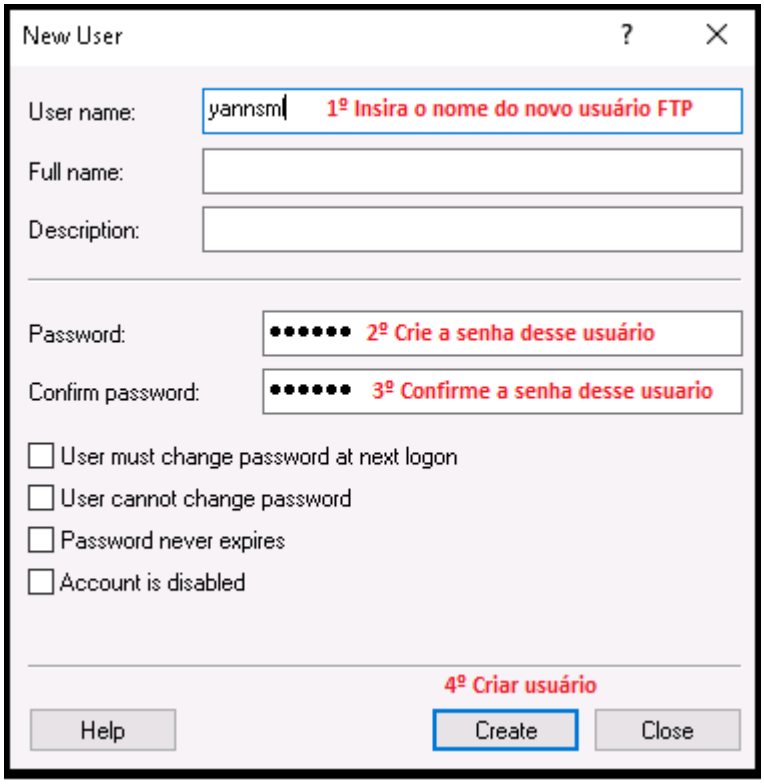
Figura 63 – Criação Grupo FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Ainda na opção “Usuários e Grupos local” foi acessado com o botão direito a aba “Usuários” e foi selecionada a opção “Novo Usuário”. Conforme mostrado na Figura 64, o novo usuário foi nomeado como “yannsml” e foi informada uma senha e sua confirmação. Então, foi criado o usuário.

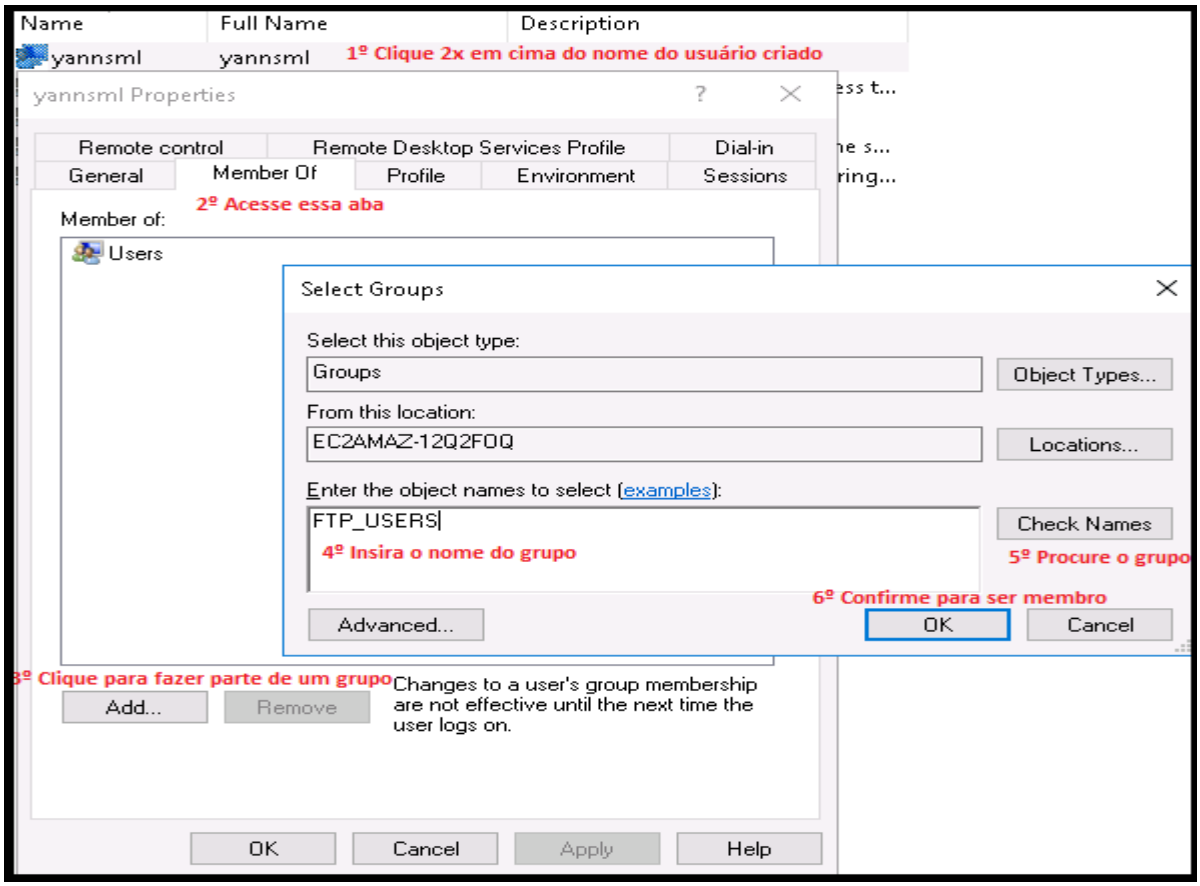
Figura 64 – Criação usuário FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Foi acessada a aba “Usuários” e conforme apresentado na Figura 65, foi selecionado o usuário criado anteriormente utilizando o botão direito e selecionado a opção “Propriedades”. Foi acessada a aba “Membro de”, selecionada a opção “Adicionar” e inserido o nome do grupo criado anteriormente para que esse usuário tornasse um membro desse grupo. Foi feito o mesmo procedimento da Figura 64 e 65 para a criação do usuário “gabryelsml”.

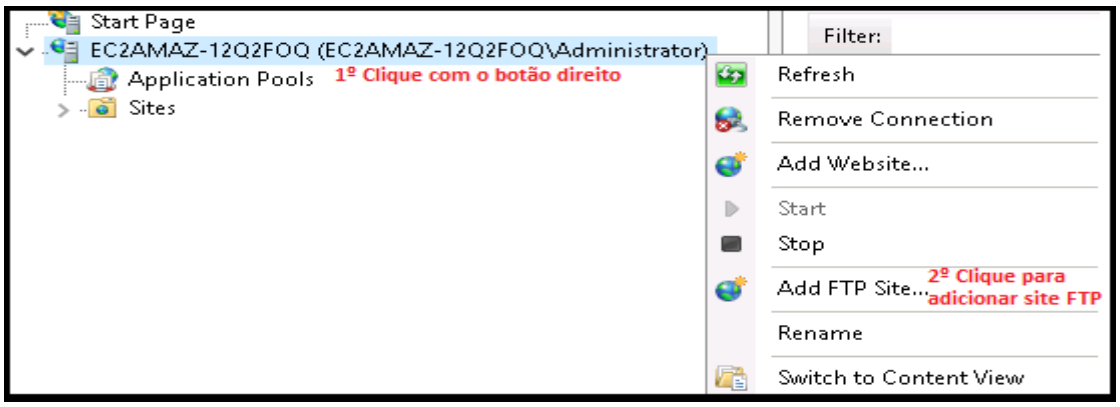
Figura 65 – Associação usuário ao grupo FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Foi acessado o aplicativo “Gerenciamento de Servidor” e foi selecionada a opção “Gerenciamento de Serviços de Informação da *Internet”.* Conforme ilustrado na Figura 66, foi selecionado, com o botão direito, a máquina que foi instalada os recursos *Web* e foi acessada a opção de adicionar novo site FTP.

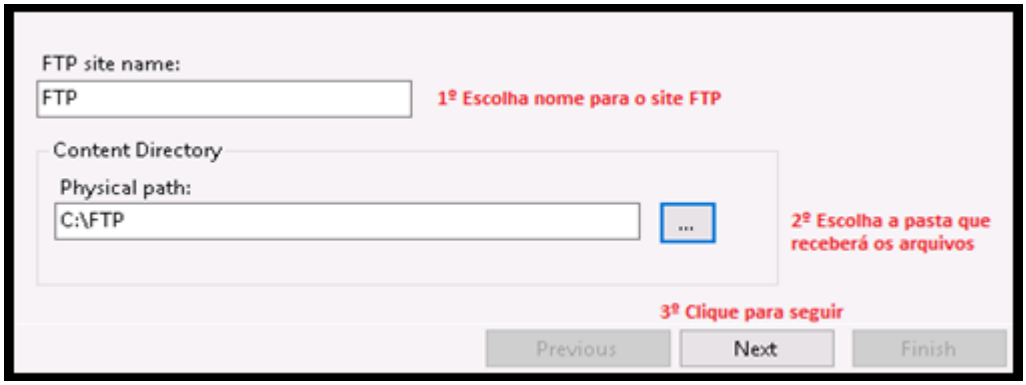
Figura 66 – Selecionar criação novo site FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Conforme mostrado na Figura 67, foi escolhido um nome para o site FTP e a pasta que foi armazenado os arquivos FTP.

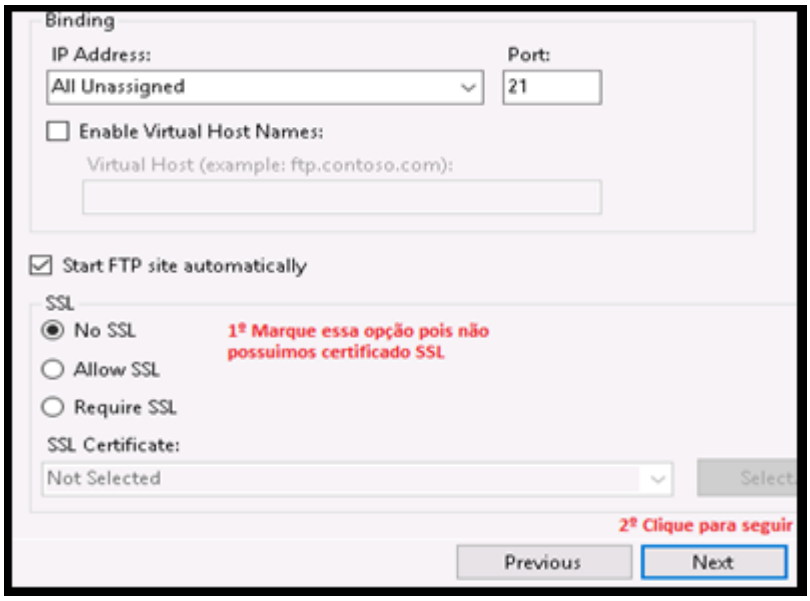
Figura 67 – Nomeação servidor FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

A Figura 68 mostra todos os IP que podem acessar o servidor FTP e que não é necessário utilização de certificado SSL.

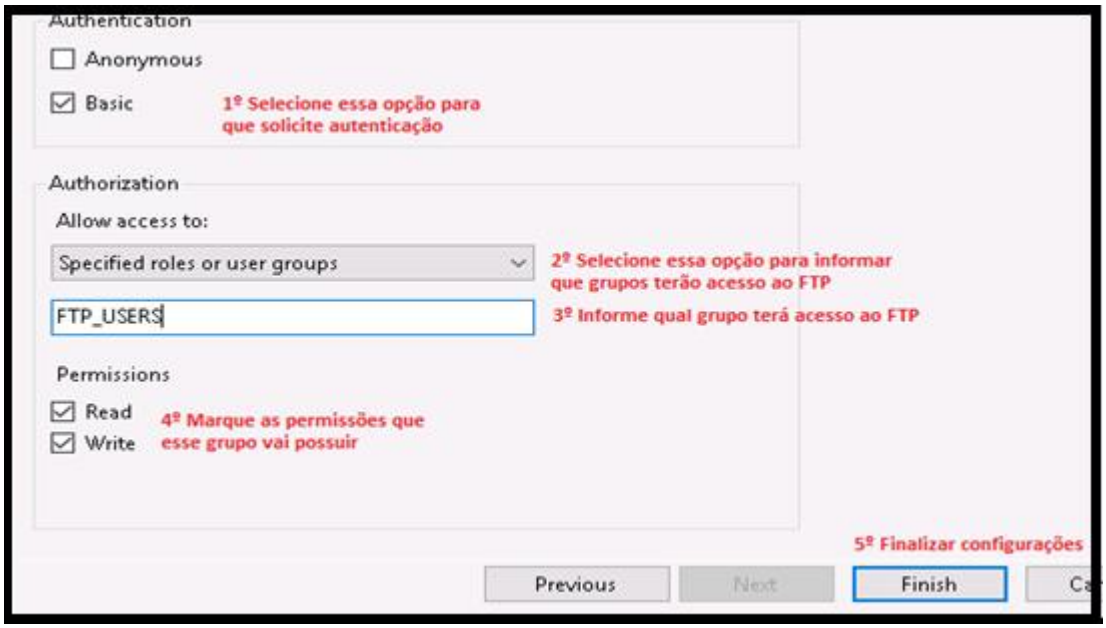
Figura 68 – Seleção de IP e SSL FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

A Figura 69 mostra que é necessário autenticação para acesso ao servidor FTP, sendo informado o grupo que contém os usuários que utilizam o servidor FTP. Estes usuários possuem permissão para ler e escrever arquivos desse servidor e, então, é criado esse site FTP.

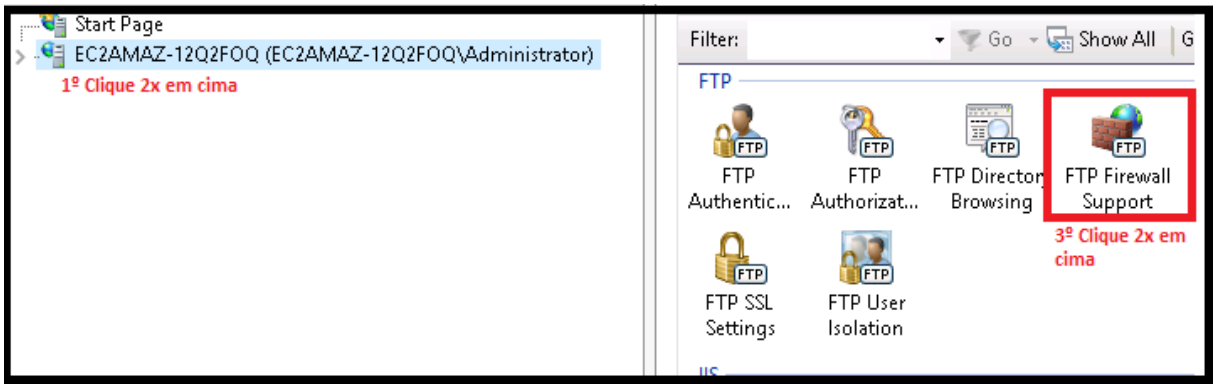
Figura 69 – Seleção autenticação e grupos FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

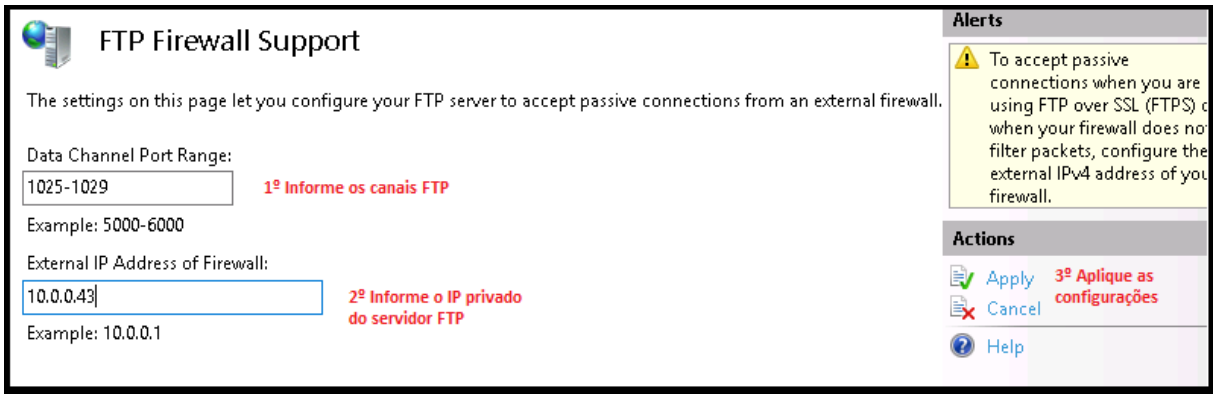
Conforme mostrado na Figura 70, foi selecionada a máquina em que foi instalada os recursos *Web* e acessado o ícone “Suporte *Firewall* FTP”. Nessa opção, como mostrado na Figura 71, foram informados os canais que o servidor FTP utilizará e o IP desse servidor. Então, foi aplicada as configurações. Por fim, selecionando a máquina, foi acessada a opção para reiniciar o servidor.

Figura 70 – Acesso ao Suporte *Firewall* FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

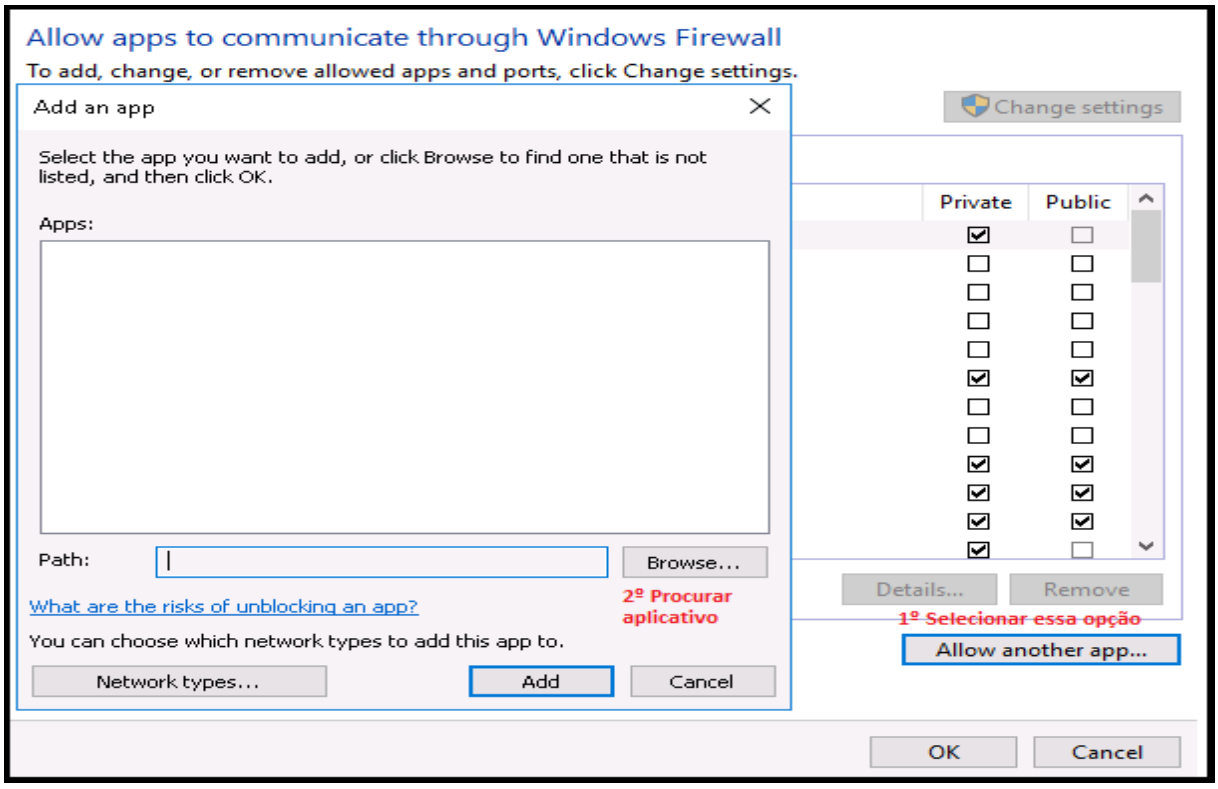
Figura 71 - Informar canais FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Foi aberto o “Painel de Controle” do *Windows Server 2016*, sendo acessada a opção “Sistema e Segurança” e a aba “Permitir um aplicativo através do *Firewall* do *Windows*”. Nessa janela, como ilustrado na Figura 72, foi selecionada a opção “Permitir outro aplicativo” e, então, foi selecionada a opção “Procurar”. Na busca foi selecionado o arquivo “svchost.exe”, que foi adicionado ao *Firewall*.

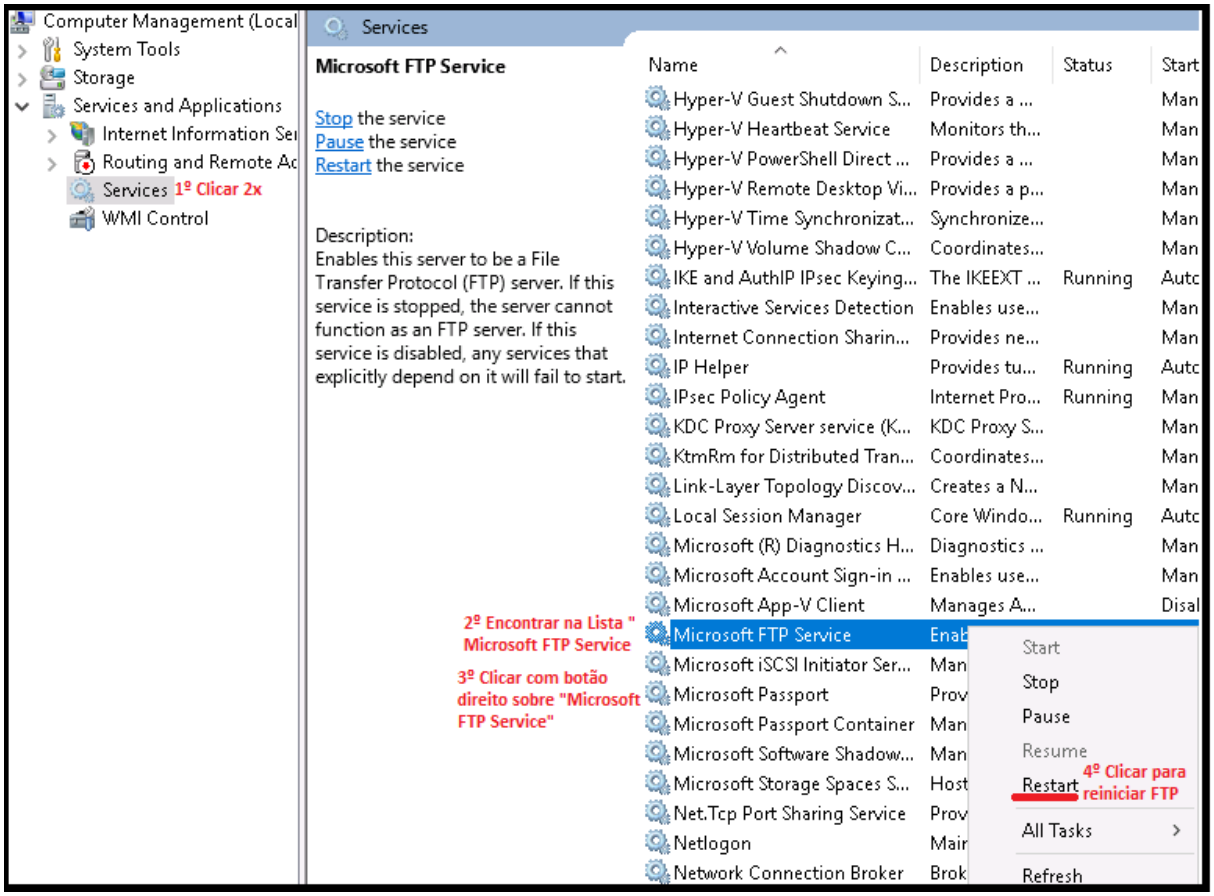
Figura 72 - Procurar novo aplicativo *Firewall* FTP



Fonte: alterado baseado em Windows Server 2016 (2020)

Foi aberto o aplicativo “Gerenciamento de computadores” e conforme mostrado na Figura 73, foi expandida a aba “Serviços e aplicações”, selecionada a aba “Serviços”, acessado o serviço “*Microsoft* FTP *Service*”, utilizando o botão direito e acessada a opção “Reiniciar”, para reiniciar o serviço FTP.

Figura 73 – Reiniciar serviço FTP *Windows*



Fonte: alterado baseado em Windows Server 2016 (2020)

Foi acessado o aplicativo “Gerenciamento de Servidor” e a opção “Gerenciamento de Serviços de Informação da *Internet”*, sendo selecionada a máquina que foi instalada os recursos *Web* e foi reiniciado o servidor FTP.

4.3.4 Servidor *Web*

O servidor *Web* foi criado para acessar via navegador o servidor FTP, possibilitando o *download* de seus arquivos. Esse servidor possui uma página HTML que apresenta um formulário para preencher o usuário e senha para tentar se autenticar no servidor FTP. Esse usuário e senha são os mesmos criados no servidor FTP.

Por trás dessa página HTML, existe uma página PHP que vai receber esses dados do HTML, se conectar ao banco de dados RDS e verificar se os parâmetros passados são verdadeiros. Se os parâmetros forem verdadeiros, então será redirecionado para a página do servidor FTP.

Para a criação da instância desse servidor, foram usados os passos mostrados no anexo D – com a AMI Ubuntu 18.04 e escolhendo a DMZ “DMZ – Web” – e foi escolhido o mesmo par de chaves criado na criação do servidor VPN. Para acessar essa instância, foram utilizados os passos mostrados no anexo B.

Como esse servidor se encontra na sub-rede privada e é necessário baixar o *apache2*, *mysql-server* e o php, foram utilizados os passos mostrados no anexo A. Assim, essa instância teve acesso à *Internet*. Os comandos usados para baixar esses programas foram “sudo su”, “*apt-get update*”, “*apt-get install apache2*”, “*apt-get install mysql-server*” e “*apt-get install php libapache2-mod-php php-mysql*”, nessa ordem. Ao finalizar os *downloads,* foi retirado o acesso à *Internet,* utilizando passos mostrados no anexo A. O arquivo “index.html”, encontrado no caminho “/var/www/html”, foi alterado para ficar de acordo com a Figura 74.

Figura 74 – Códigos HTML

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
        <title>Formulário</title>
    </head>
    <body>
        <form name="f_login" method="post" action="post.php">
            <label>Usuário: </label><br/>
            <input type="text" name="f_usuario" size="40" maxlength="30"/><br/><br/>
            <label>Senha: </label><br/>
            <input type="password" name="f_senha" size="40" maxlength="20"/><br/><br/>

            <input type="submit" value="Enviar"/>
            <input type="reset" value="Limpar"/>
    </body>
</html>
```

Fonte: alterado baseado em PuTTY (2021)

Na Figura 74 é ilustrado:

- Linha 1: Início das instruções HTML.
- Linha 2: Início das instruções do cabeçalho do HTML.
- Linha 3: É especificado a codificação dos caracteres.
- Linha 4: Informando que o título desse HTML é “Formulário”.
- Linha 5: Fim das instruções do cabeçalho do HTML.
- Linha 6: Início das instruções do corpo do HTML.
- Linha 7:Informando que o nome do formulário é “f_*login*”, utiliza o método “post” e vai enviar os dados para o arquivo “post.php”.
- Linha 8: Rótulo da variável do usuário ftp.
- Linha 9: Variável que vai receber o usuário ftp.
- Linha 10:Rótulo da variável da senha ftp.
- Linha 11: Variável que vai receber a senha ftp.
- Linha 12: Botão para enviar o conteúdo das variáveis usuário e senha para o arquivo post.php.
- Linha 13: Botão para limpar o conteúdo das variáveis usuário e senha.
- Linha 14: Fim das instruções do corpo do HTML.
- Linha 15: Fim das instruções HTML.

Para criação da tabela que armazenará o usuário e senha dos clientes FTP no banco de dados, foi criado um arquivo no caminho “/var/www/html/” chamado “*createtable*.php” com os códigos mostrados na Figura 75 e, então, executado.

Figura 75 – Códigos *createtable*.php

```php
<?php
    define('host','db.redestcc.tk');
    define('usuario','db');
    define('senha','masterkey');
    define('banco','db');

    $conexao = mysqli_connect(host,usuario,senha,banco) or die ('Não foi possivel conectar com o banco');

    $query="CREATE TABLE controlador (usuario varchar(50) NOT NULL, senha varchar(50) NOT NULL)";
    $result=mysqli_query($conexao,$query);

?>
```

Fonte: alterado baseado em PuTTY (2021)

Na Figura 75 é mostrado:

- Linha 1: Início do programa PHP.
- Linha 2: Criando uma constante chamada “host*”* com valor “db.redestcc.tk”.

- Linha 3: Criando uma constante chamada “usuário” com valor “db”.
- Linha 4: Criando uma constante chamada “senha” com valor “*masterkey*”.
- Linha 5: Criando uma constante chamada “banco” com valor “db”.
- Linha 6: Variável que vai receber a conexão com o banco RDS, se não for possível a conexão, será retornado a mensagem “Não foi possível conectar com o banco”.
- Linha 7: *Query* da criação de uma tabela, a tabela foi chamada de “controlador” e possui as colunas usuário e senha, no qual eles são atributos *varchar,* de tamanho 50.
- Linha 8:Executando a *query*.
- Linha 11: Fim do programa PHP.

Para inserir os dados dos usuários ftp, foi criado o arquivo “*insert*.php” no caminho “/var/www/html/”. Os códigos inseridos nesse arquivo foram ilustrados na Figura 76.

Figura 76 – Códigos *insert*.php

```php
<?php
    define('host','db.redestcc.tk');
    define('usuario','db');
    define('senha','masterkey');
    define('banco','db');

    $conexao = mysqli_connect(host,usuario,senha,banco) or die ('Não foi possivel conectar com o banco');

    $query="insert into controlador (usuario,senha) values ('yannsml',md5('YANN123SML!'))";
    $result=mysqli_query($conexao,$query);

    $query="insert into controlador (usuario,senha) values ('gabryelsml',md5('YANN123SML!'))";
    $result=mysqli_query($conexao,$query);

?>
```

Fonte: alterado baseado em PuTTY (2021)

Esse código foi executado e a Figura 76 é mostra:

- Linha 1: Início do programa PHP.
- Linha 2: Criando uma constante chamada “host*”* com valor “db.redestcc.tk”.
- Linha 3: Criando uma constante chamada “usuário” com valor “db”.
- Linha 4: Criando uma constante chamada “senha” com valor “*masterkey*”.
- Linha 5: Criando uma constante chamada “banco” com valor “db”.

- Linha 6: Variável que vai receber a conexão com o banco RDS, se não for possível a conexão, será retornado a mensagem; “Não foi possível conectar com o banco”.
- Linha 7: *Query* para inserir os valores usuário: yannsml e senha:YANN123SML! na tabela controlador. A senha é encriptada, usando o algoritmo MD5.
- Linha 8: Executando a *query.*
- Linha 9: Sobrepondo os valores da *query* anterior, essa *query* vai inserir os valores usuário: gabryelsml e senha:YANN123SML! na tabela controlador. A senha é encriptada, usando o algoritmo MD5.
- Linha 10: Executando a nova *query*.
- Linha 11: Fim do programa PHP.

Para verificar se os dados informados no formulário HTML estão corretos e para redirecionar a página para o servidor FTP, foi criado um arquivo chamado “*post*.php” no caminho “/var/www/html/”. Os códigos inseridos nesse arquivo foram ilustrados na Figura 84.

Figura 77 – Códigos *post*.php

```php
<?php
    define('host','db.redestcc.tk');
    define('usuario','db');
    define('senha','masterkey');
    define('banco','db');

    $conexao = mysqli_connect(host,usuario,senha,banco) or die ('Não foi possivel conectar com o banco');

    $usuario=mysqli_real_escape_string($conexao,$_POST["f_usuario"]);
    $senha=mysqli_real_escape_string($conexao,$_POST["f_senha"]);

    $query="select * from controlador where usuario = '{$usuario}' and senha = md5('{$senha}')";
    $result=mysqli_query($conexao,$query);
    $linhas=mysqlI_num_rows($result);

    if($linhas==1){
        $site="ftp://".$usuario.":".$senha."@ftp.redestcc.tk";
        header("Location: ".$site);
        exit();
    }
    else{
        header('Location: index.html');
        exit();
    }

?>
```

Fonte: alterado baseado em PuTTY (2021)

Na Figura 77 é mostrado:

- Linha 1: Início do programa PHP.
- Linha 2: Criando uma constante chamada “host*”* com valor “db.redestcc.tk”.
- Linha 3: Criando uma constante chamada “usuário” com valor “db”.
- Linha 4: Criando uma constante chamada “senha” com valor “*masterkey*”.
- Linha 5: Criando uma constante chamada “banco” com valor “db”.
- Linha 6: Variável que vai receber a conexão com o banco RDS, se não for possível a conexão, será retornado a mensagem “Não foi possível conectar com o banco”.
- Linha 7: Recebendo via post o usuário informado na página HTML.
- Linha 8: Recebendo via post a senha informada na página HTML.
- Linha 9: *Query* usada para procurar no banco o usuário e senha informados.
- Linha 8: Executa a *query*.
- Linha 9: Armazena a quantidade de linhas retornadas ao executar a query.
- Linha 10: Verifica se a quantidade de linha retornadas é 1.
- Linha 11: Se a condição da linha 10 for verdade, então ele monta a URL do servidor FTP. A Url é montada iniciando com “ftp://”, depois o nome do usuário ftp, a seguir a senha, por fim o caractere “@” mais o *hostname* do servidor FTP. Por exemplo: “ftp://yannsml:YANN123SML!@ftp.redestcc.tk”.
- Linha 12: Redireciona a página para a Url criada na linha 11.
- Linha 13: Encerra o programa PHP.
- Linha 14: Entra nessa condição se a linha 10 não for verdadeira.
- Linha 15: Redireciona novamente para a página HTML.
- Linha 16 e 17: Fim do programa PHP.

Ao fim foi utilizado o comando “*service apache2 restart”* para reiniciar o serviço *web*.

4.3.5 *Workspaces*

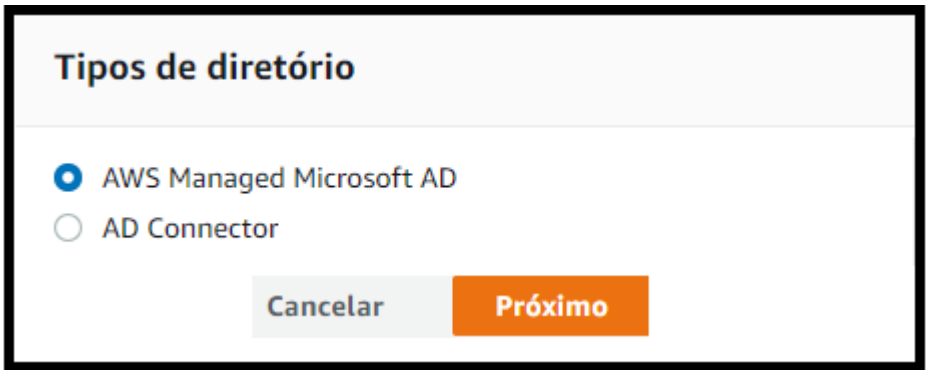
Os *Workspaces* são serviços de implementação de *desktops* virtualizados. Eles foram criados para que os clientes possam ter acesso aos servidores da Rede VPC.

Para ser possível a criação dos *desktops* virtuais é necessária a criação de um *active directory*. Essa funcionalidade possibilita que cargas de trabalho e recursos

AWS que tenham reconhecimento de diretório utilizem o *Active Directory*, que é gerenciado na AWS. Com esse *Active Directory* é possível associar instâncias EC2 e bancos de dados RDS em um domínio, além de utilizar o *Amazon Workspaces* com usuários e grupos desse *Active Directory* (AWS, 2021).

Foi acessado o serviço "*WorkSpaces*" da AWS e escolhida a opção "*Directories". Depois* acessou-se a funcionalidade "*Set up Directory"*. Conforme mostrado na Figura 78, foi marcada a opção "AWS *Managed Microsoft* AD".

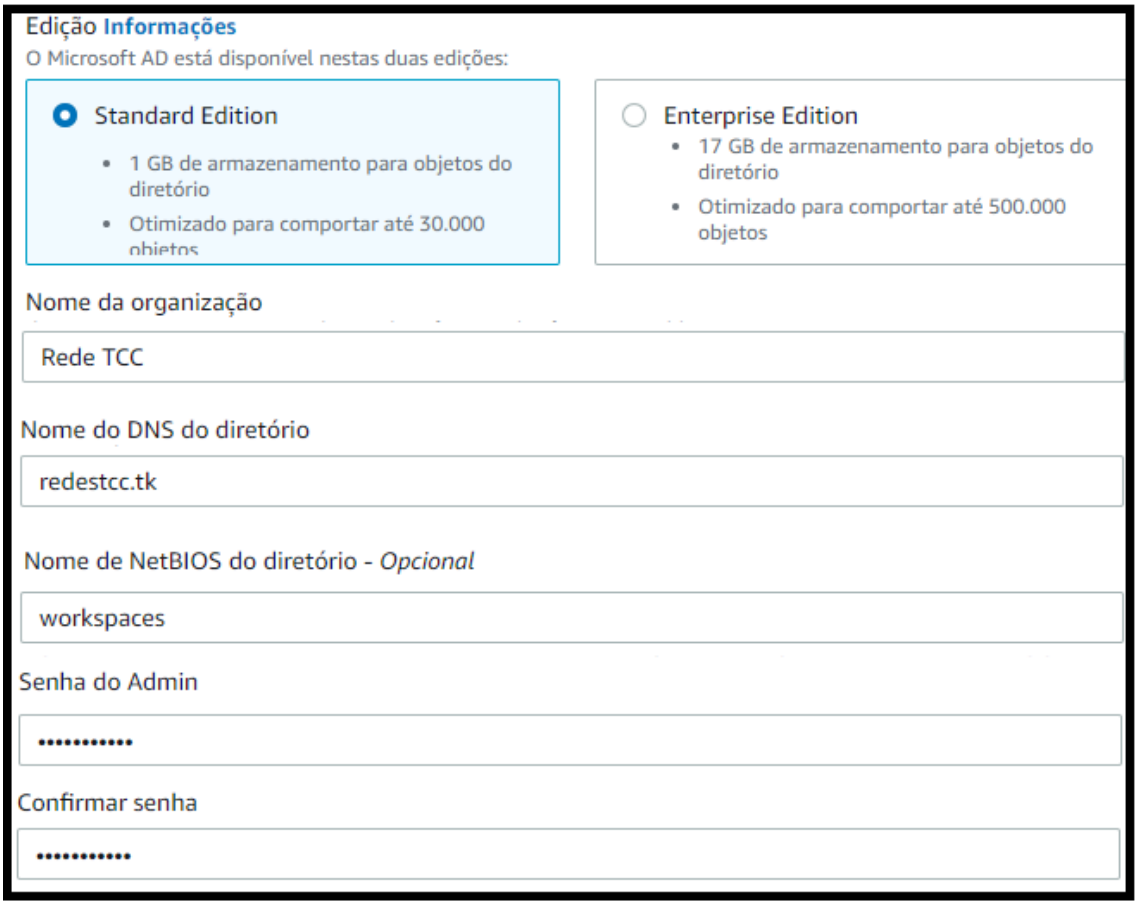
Figura 78 – Escolha do tipo de diretório



Fonte: alterado baseado em AWS (2021)

Como mostrado na Figura 79, foi escolhida a edição "*Standart Edition"*. Essa opção possui 1 GB de armazenamento e pode comportar até 30.000 objetos. Além disso, foi informado que o nome da organização é "Rede TCC", o domínio criado possui o nome de "redestcc.tk", informado que o identificador do domínio é "*workspaces*" e foi escolhido uma senha, sendo a mesma confirmada.

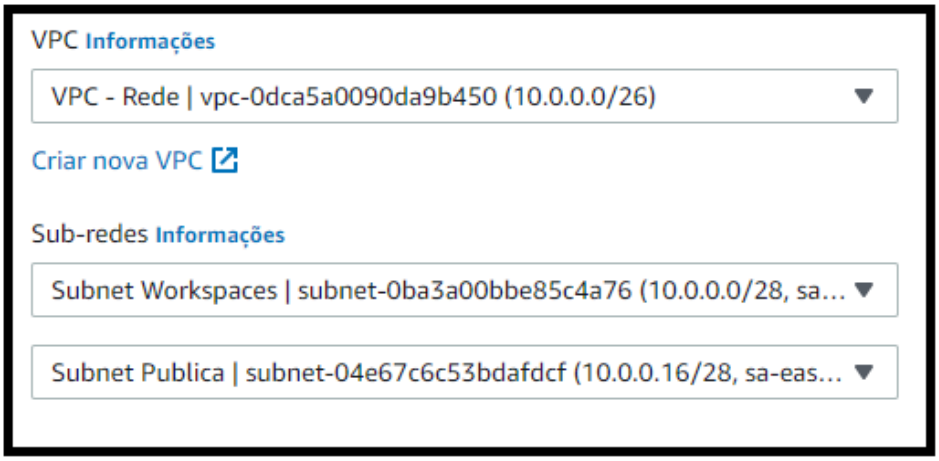
Figura 79 – Escolha da edição de diretório



Fonte: alterado baseado em AWS (2021)

Como ilustrado na Figura 80, foi definido que este diretório fará parte da VPC-Rede e das sub-redes *Workspaces* e publica.

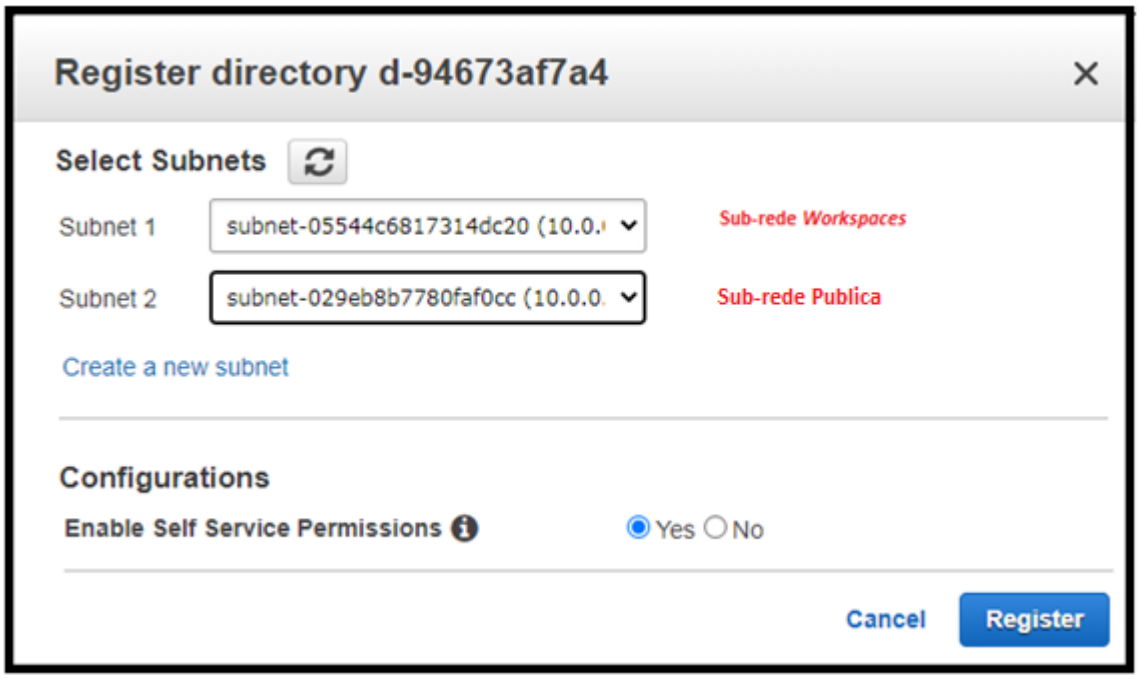
Figura 80 – Escolha da rede do diretório



Fonte: alterado baseado em AWS (2021)

A seguir foi confirmada as informações de criação desse diretório e, então, ele foi criado. Após a criação, foi selecionado o diretório e escolhida a aba "*Actions*" e a opção "*Register*". Como ilustrado na Figura 81, foi definido que este diretório foi registrado na sub-rede *Workspace* e na sub-rede pública.

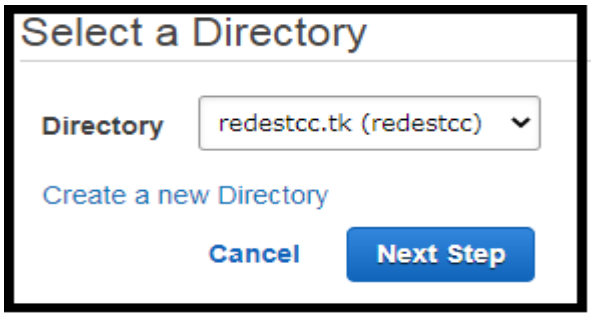
Figura 81 – Registrando diretório nas sub-redes



Fonte: alterado baseado em AWS (2021)

Na página "*Workspaces*", foi escolhida a opção "*WorkSpaces"* e a opção "*Launch WorkSpaces*". Como ilustrado na Figura 82, foi definido o diretório criado anteriormente para ser o diretório desse *Workspaces*.

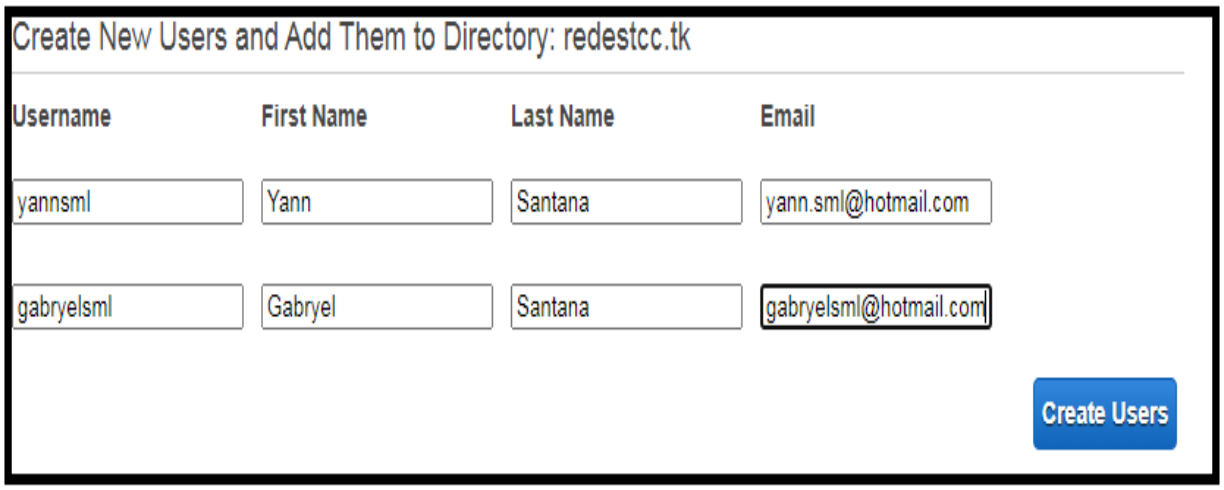
Figura 82 – Escolha do Diretório dos *Workspaces*



Fonte: alterado baseado em AWS (2021)

Como ilustrado na Figura 83, foram criados dois usuários (yannsml e gabryelsml).

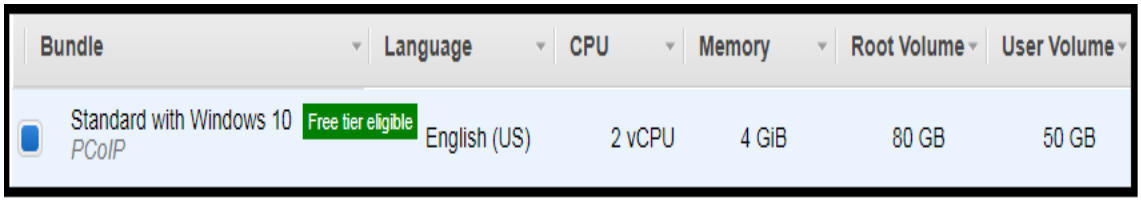
Figura 83 - Criação usuários *Workspaces*



Fonte: alterado baseado em AWS (2021)

Como mostrado na Figura 84, foi escolhida uma instância que possui *Windows* 10, na linguagem inglês, duas CPUs, 4 GB de memória RAM, 80 GB de armazenamento root e 50 GB de armazenamento para o usuário.

Figura 84 – Instância *Workspaces*



Fonte: alterado baseado em AWS (2021)

O modo de execução escolhido foi o *AutoStop* e não foi utilizado encriptação no armazenamento. Por fim, houve verificação das informações usadas para a criação desses *WorkSpaces* e, então, ele foi criado.

5 ANÁLISE DOS RESULTADOS OBTIDOS

Esse capítulo mostra como foram realizados e analisados os testes de usabilidade, confiabilidade e desempenho, apresentando telas evidenciando os resultados.

5.1 Teste e análise de usabilidade

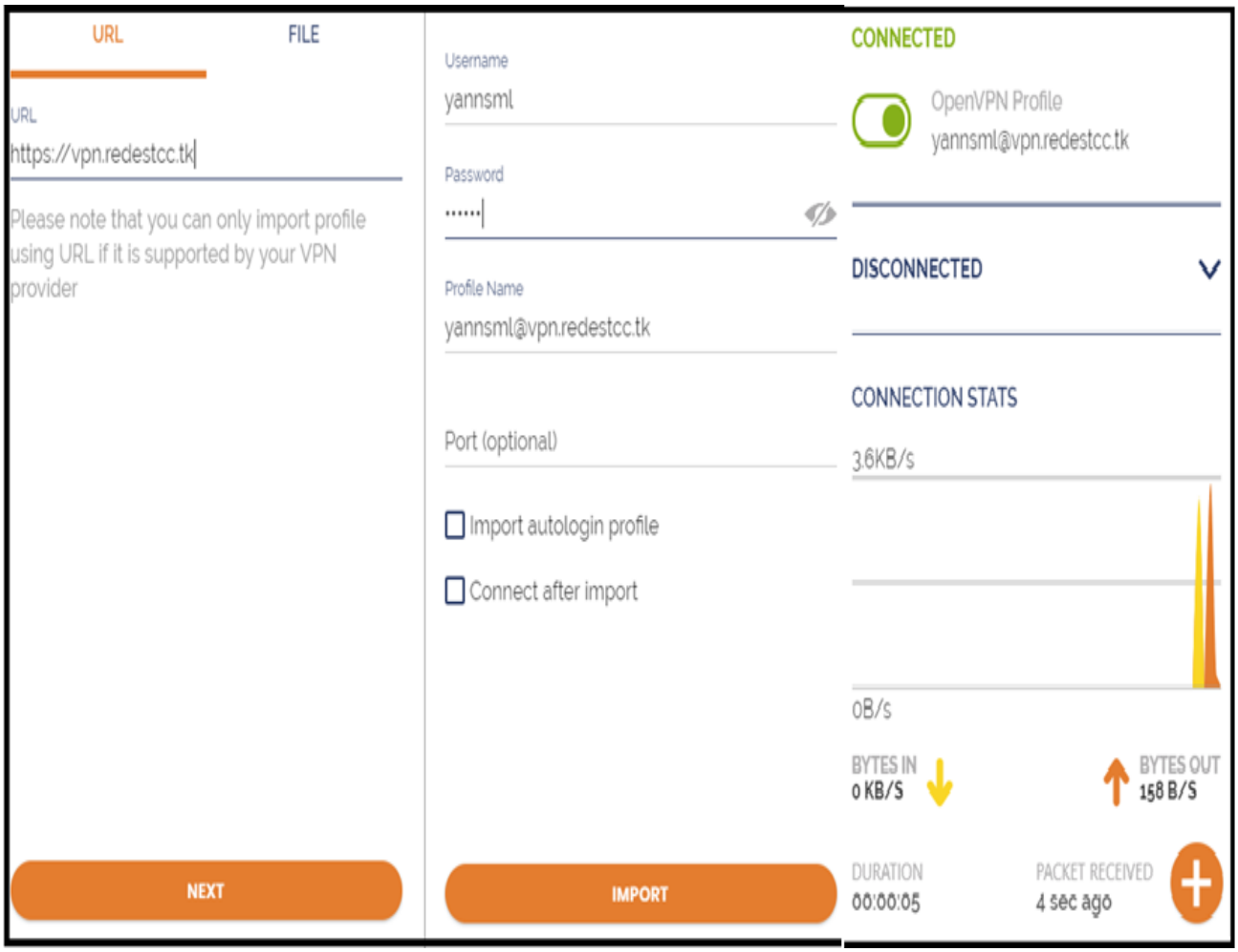
Para teste de usabilidade, foi testada a conectividade dos clientes com o *Workspaces* e servidor VPN com os servidores da sub-rede privada. Além disso, foram testados *downloads*, *uploads*, renomeação e exclusão de arquivos armazenados no servidor FTP. Também foi feito teste autenticando no servidor *Web Proxy* para verificar se o IP público apresentado era o IP do servidor *Web Proxy.*

5.1.1 Teste e análise de usabilidade do servidor VPN

Neste subtópico foram feitos testes de conexão com o servidor VPN, verificando se os usuários cadastrados conseguem logar no servidor VPN e se esses usuários conseguem acessar os serviços dos servidores FTP e *Web*.

Conforme ilustrado na Figura 85, para conectar ao servidor foi aberto o aplicativo *OpenVPN*. Para logar é necessário memorizar o usuário no aplicativo *OpenVPN*. Para isso, foi informado o nome “vpn.redestcc.tk” que roteia para o IP público do servidor VPN. Na aba seguinte foi informado um usuário que foi criado anteriormente na criação do servidor VPN, além de informar a sua senha. A seguir, foi logado utilizando esse usuário e informando a senha. Por fim, usuário foi logado na VPN.

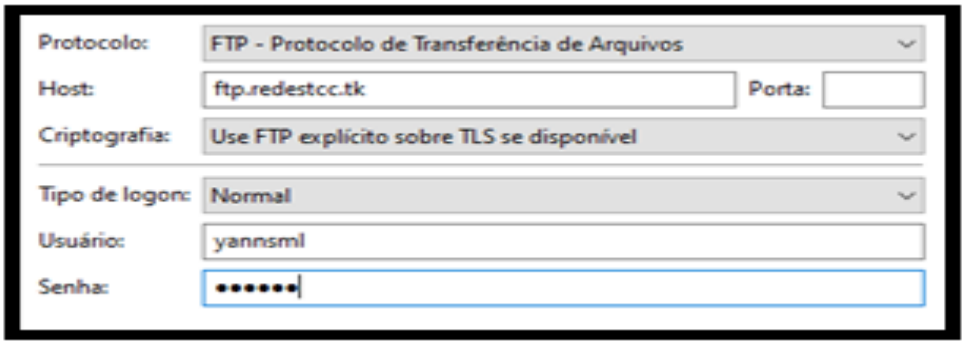
Figura 85 - Memorizar usuário yannsml *OpenVPN*



Fonte: alterado baseado em OpenVPN (2021)

Logado com esse usuário no *OpenVPN*, foi aberto o *Filezilla* e inserido as informações apresentadas na Figura 86. Foi inserido o *hostname* do servidor FTP, o usuário “yannsml” e sua senha.

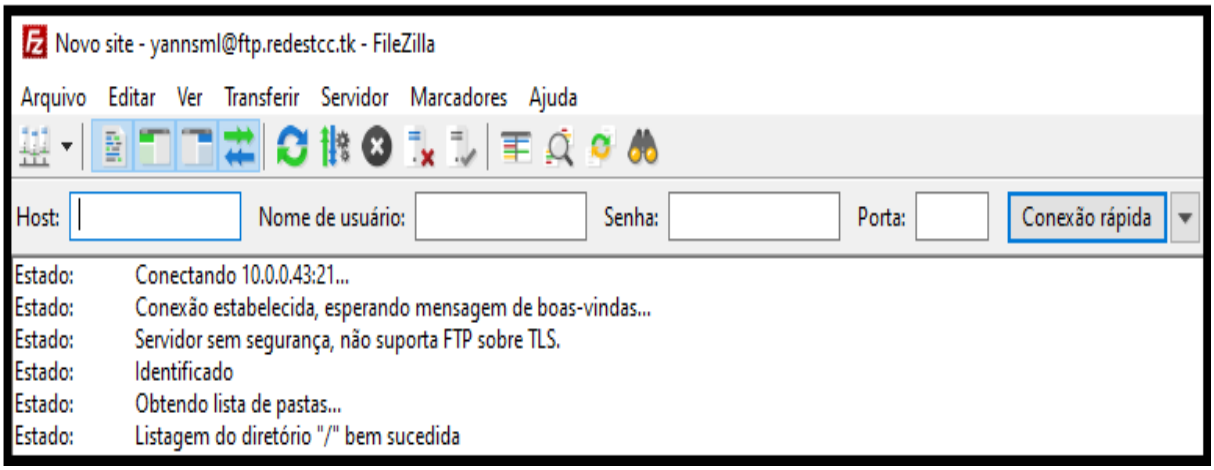
Figura 86 – *User* VPN yannsml informando credenciais servidor FTP



Fonte: alterado baseado em FileZilla (2021)

Como ilustrado na Figura 87, a conexão com o servidor FTP foi bem-sucedida.

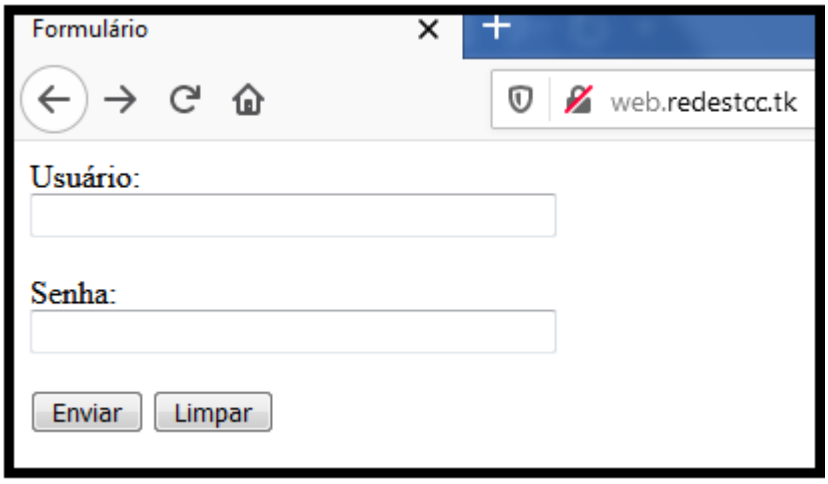
Figura 87 – *User* VPN yannsml *success auth* servidor FTP



Fonte: alterado baseado em FileZilla (2021)

Com o usuário “yannsml” logado no *OpenVPN* e autenticado no servidor *Web Proxy*, foi aberto o navegador *Firefox* e solicitado conexão com a URL “web.redestcc.tk”, e como ilustrado na Figura 88, a conexão foi bem-sucedida.

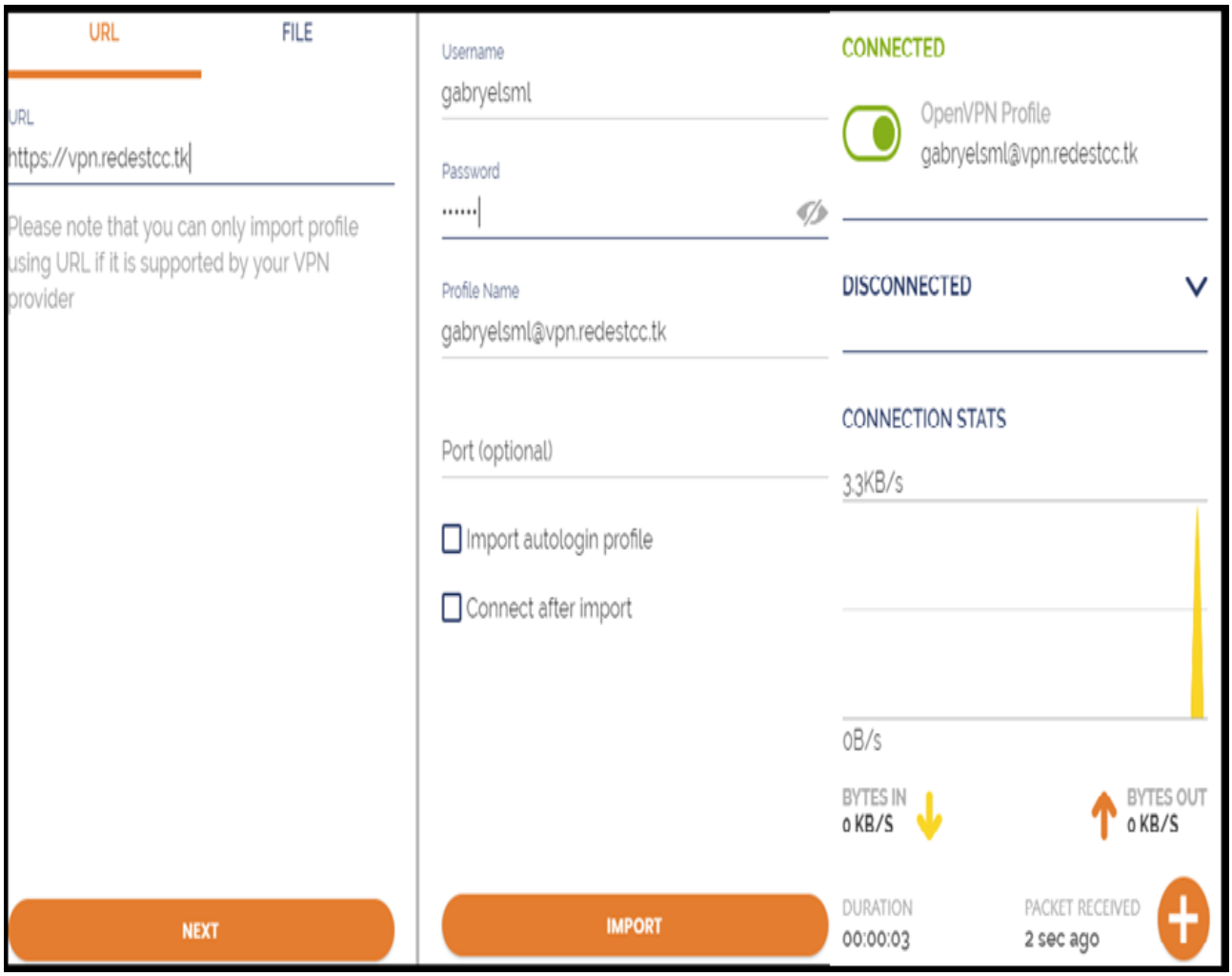
Figura 88 – *User* VPN yannsml *success auth* servidor *Web*



Fonte: alterado baseado em Firefox (2021)

Conforme apresentado na Figura 89, foi informado o nome “vpn.redestcc.tk. Na aba seguinte, foi o *login* do outro usuário - criado na criação do servidor VPN - e sua senha para assim memorizar esse usuário no aplicativo da *OpenVPN*. A seguir, foi feito *login* e então é apresentado que a conexão foi bem-sucedida.

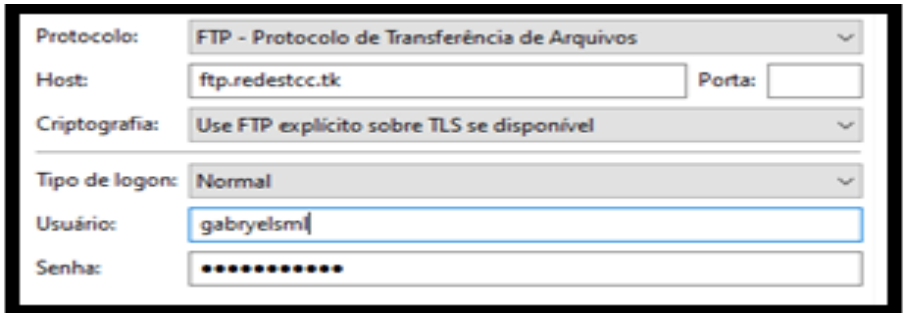
Figura 89 - Memorizar usuário gabryelsml VPN



Fonte: alterado baseado em OpenVPN (2021)

Com o usuário “gabryelsml” logado no *OpenVPN*, foi aberto o *Filezilla* e inserido as informações apresentadas na Figura 90. Foi inserido o *hostname* do servidor FTP, o usuário “gabryelsml” e sua senha.

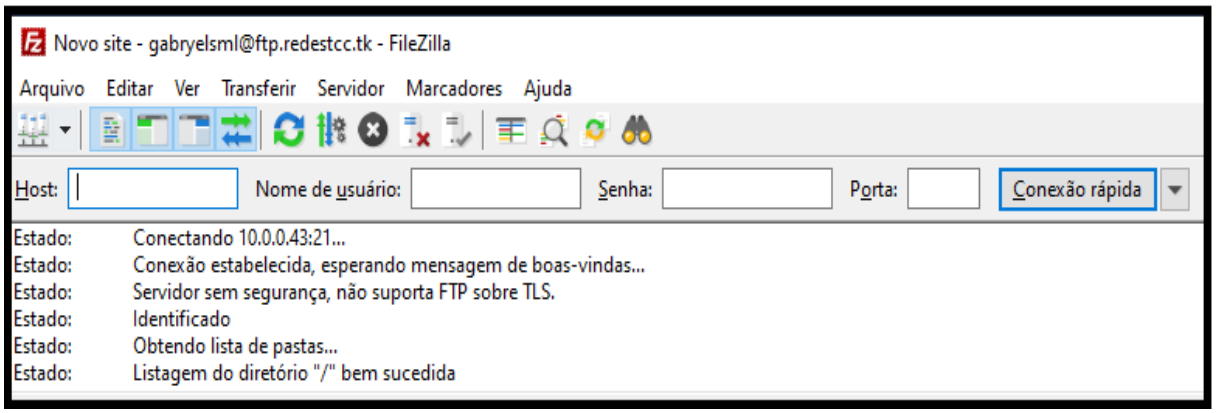
Figura 90 - *User* VPN gabryelsml informando credenciais servidor FTP



Fonte: alterado baseado em FileZilla (2021)

Conforme mostrado na Figura 91, a conexão com o servidor FTP foi bem-sucedida.

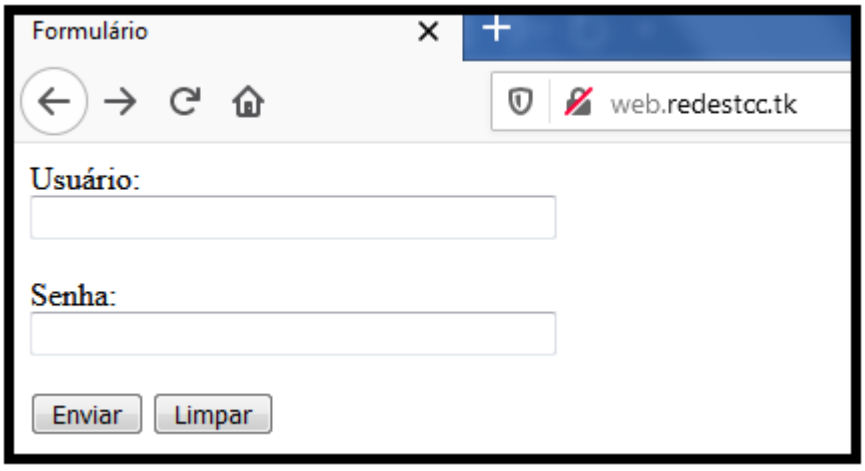
Figura 91 – *User* VPN gabryelsml *success auth* servidor FTP



Fonte: alterado baseado em FileZilla (2021)

Com o usuário “gabryelsml” logado no *OpenVPN* e autenticado no servidor *Web Proxy*, foi aberto o navegador *Firefox* e solicitado conexão com a URL “web.redestcc.tk”, e como ilustrado na Figura 92, a conexão foi bem-sucedida.

Figura 92 – *User* VPN gabryelsml *success auth* servidor *Web*



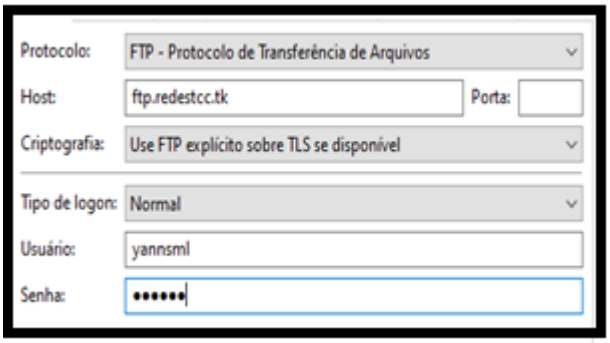
Fonte: alterado baseado em Firefox (2021)

Com esses testes foi concluído que o servidor VPN está conforme a usabilidade.

5.1.2 Teste e análise de usabilidade do servidor FTP

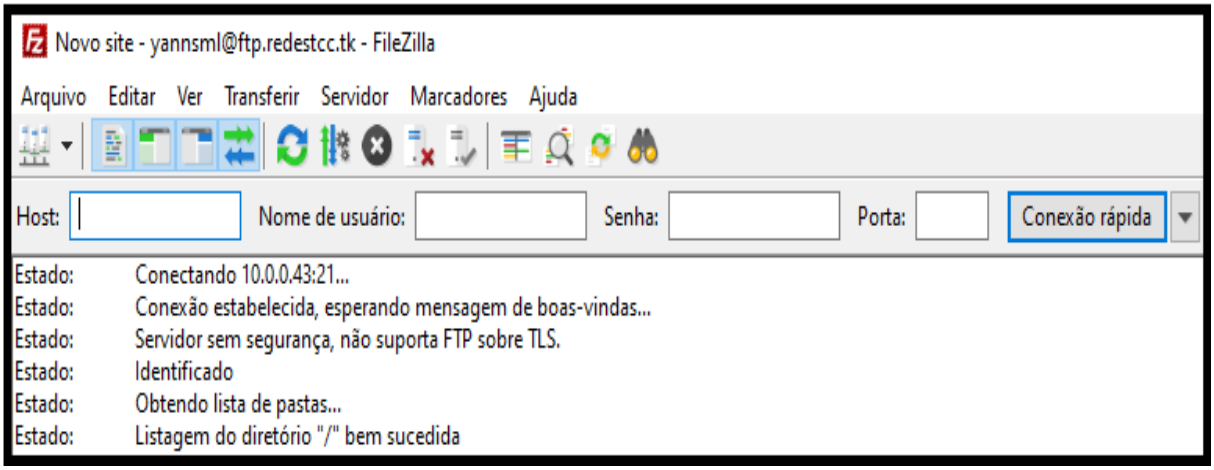
Para testar o servidor FTP, foi realizado teste de envio, recebimento, renomeação e exclusão de arquivos do servidor. Inicialmente, foi feito *login* com o usuário “yannsml” no servidor VPN. Como ilustrado na Figura 93, foi aberto o aplicativo *FileZilla* e selecionada a aba “Gerenciador de Sites”. Nessa aba, foi informado o *hostname* do servidor FTP, a porta foi deixada *Default*, foi inserido o *login* e a senha do usuário “yannsml” – criado na criação do servidor FTP – e então foi selecionada a opção “Conectar”. Conforme mostrado na Figura 94, usuário logado com sucesso.

Figura 93 - Memorização site FTP



Fonte: alterado baseado em FileZilla (2020)

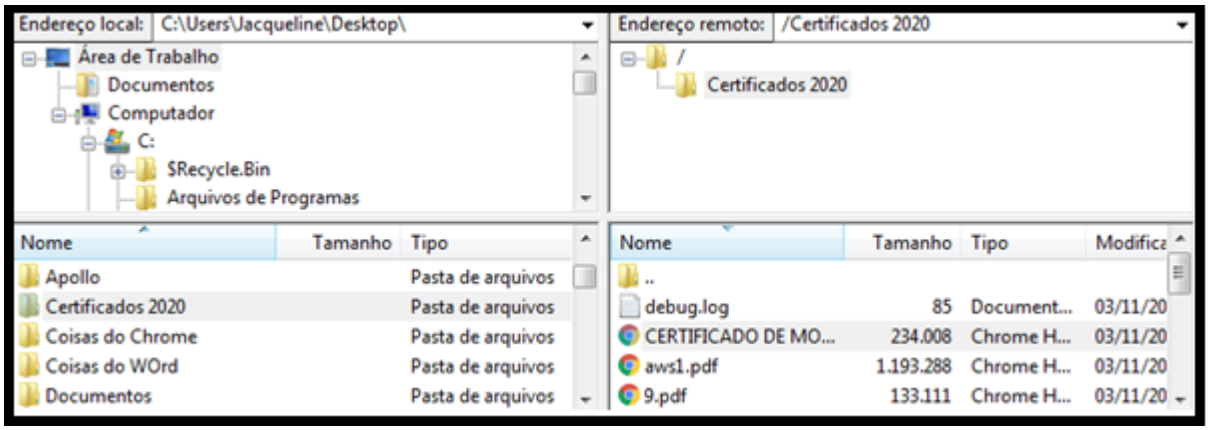
Figura 94 – Usuário logado FTP



Fonte: alterado baseado em FileZilla (2020)

Foi realizado teste de envio da pasta “Certificados 2020” e conforme ilustrado na Figura 95, a pasta foi copiada com sucesso com os documentos PDF que ela possuía no servidor FTP.

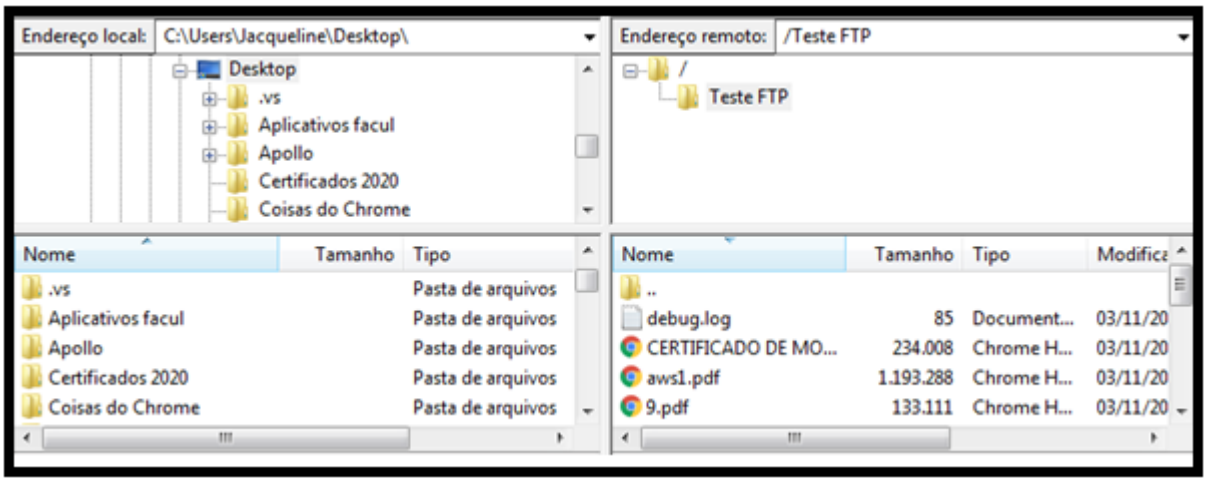
Figura 95 – Envio de arquivos servidor FTP



Fonte: alterado baseado em FileZilla (2020)

Foi realizado teste para renomear a pasta “Certificados 2020” para “Teste FTP” e como mostrado na Figura 96, a pasta foi renomeada com sucesso.

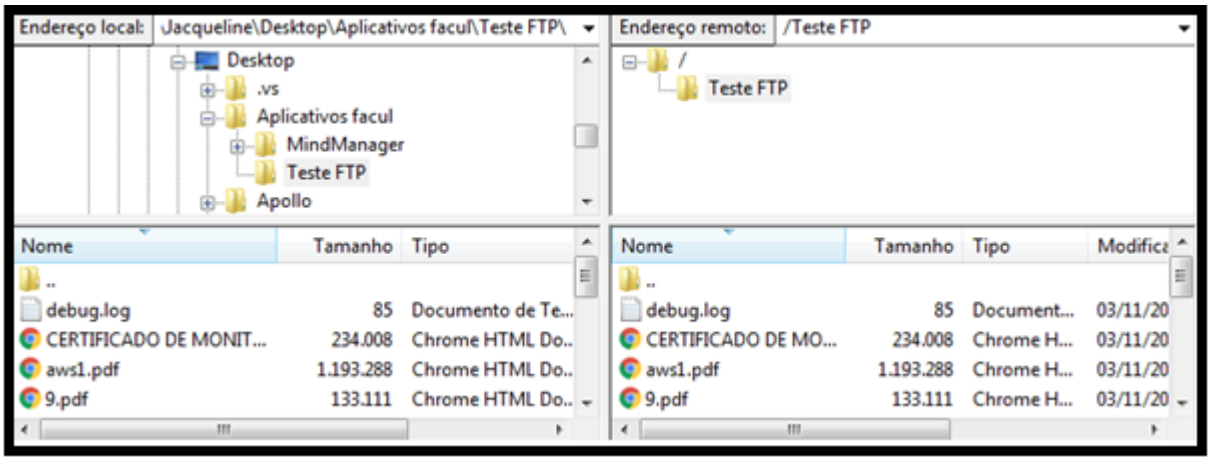
Figura 96 – Renomear pasta FTP



Fonte: alterado baseado em FileZilla (2020)

Foi feito teste baixando a pasta “Teste FTP” que está localizada no servidor FTP para a pasta “Aplicativos facul”. Conforme mostrado nas Figuras 97, a pasta foi baixada.

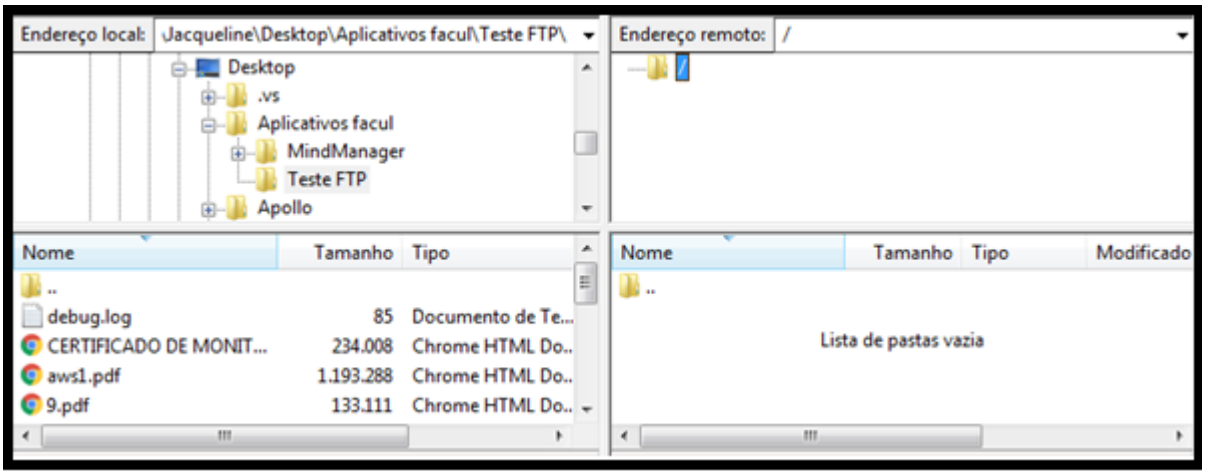
Figura 97 – Baixar pasta FTP



Fonte: alterado baseado em FileZilla (2020)

Para testar a exclusão dos arquivos no servidor FTP, foi realizada tentativa de excluir a pasta “Teste FTP”. A Figura 98 mostra que a pasta foi excluída com sucesso.

Figura 98 – Exclusão pasta FTP



Fonte: alterado baseado em FileZilla (2020)

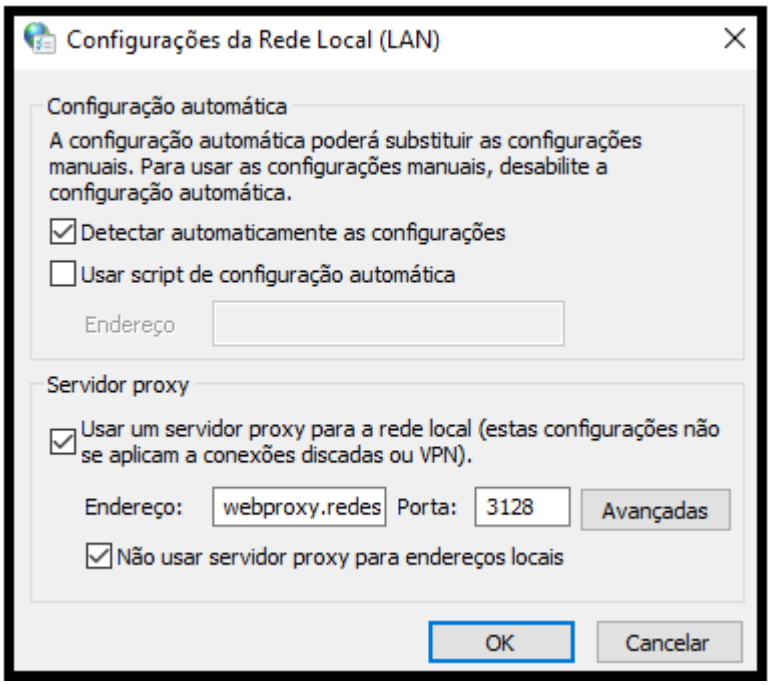
Com esses testes foi concluído que o servidor FTP está conforme a usabilidade.

5.1.3 Teste e análise de usabilidade do servidor *Web Proxy*

Para realizar teste de usabilidade do servidor *Web Proxy*, foi acessado o painel de controle do *Windows,* a opção Rede e *Internet*, depois acessado a opção Opções da *Internet*. Na aba Conexões, foi acessada a opção Configuração da LAN e então foi

informado o nome “webproxy.redestcc.tk” que roteia para o IP público do servidor *Web Proxy* e a porta de comunicação, conforme ilustrado na Figura 99.

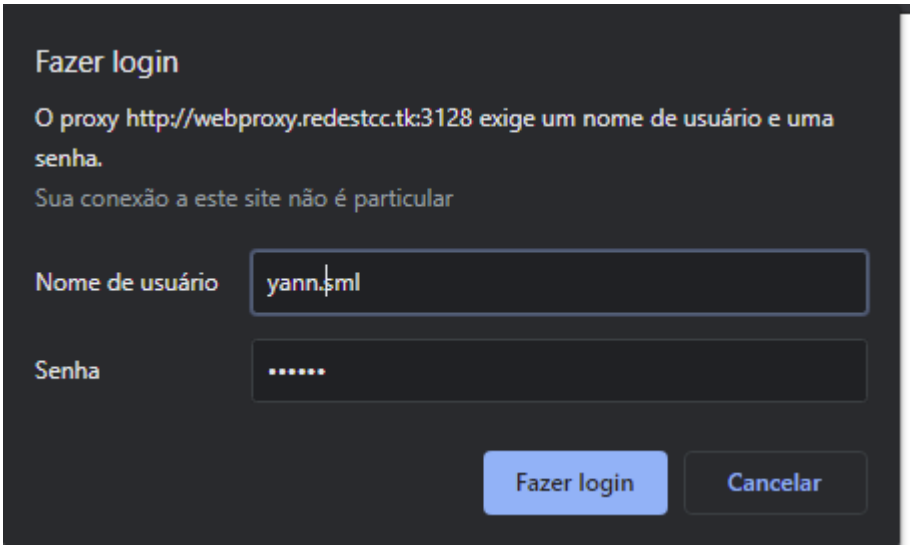
Figura 99 – Inserção *Web Proxy*



Fonte: alterado baseado em Windows 7 (2020)

Então foi aberto o navegador *Google Chrome*. Ao abrir apareceu a mensagem ilustrada na Figura 100.

Figura 100 – Solicitação de autenticação *Web Proxy*



Fonte: alterado baseado em Google Chrome (2020)

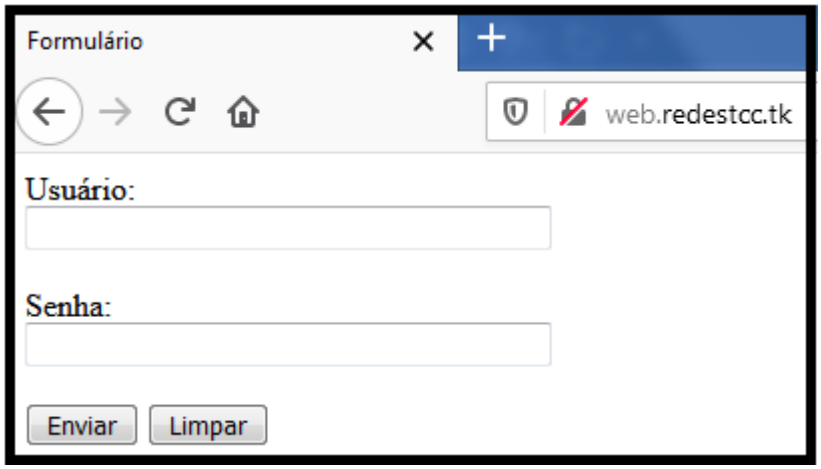
Foi informado *login "yann.sml"* e uma senha incorreta. Após esse passo, foi novamente informado a solicitação de autenticação, mostrado na Figura 100. Foi informado o *login* "YANNsml" e a senha correta, e foi mostrado novamente a Figura 100. Por fim, foi utilizado o "yann.sml" e a senha correta, e então abriu a página inicial cadastrada no *Chrome*.

A seguir, foi solicitado o acesso ao site MeuIP para verificar qual IP o cliente está utilizando. Foi verificado que o IP que o cliente está utilizando é o IP público do servidor *Web Proxy.* Portanto, concluiu-se que o servidor *Web Proxy* está cumprindo conforme a usabilidade.

5.1.4 Teste e análise de usabilidade do servidor *Web*

Para realizar o teste de usabilidade do servidor *Web* foi feito *login* no servidor VPN e a autenticação no servidor *Web Proxy*. A seguir, foi aberto o navegador *Firefox*, inserido a URL "web.redestcc.tk" e feita solicitação de conexão. Como mostrado na Figura 101, foi apresentado uma tela solicitando o usuário e senha do usuário FTP.

Figura 101 – Tela de *login* do servidor *Web*



Fonte: alterado baseado em Firefox (2021)

Foi inserido o usuário "yannsml" e sua senha. Como apresentado na Figura 102, a página foi redirecionada para a página do servidor FTP.

Figura 102 – Sucesso no *login* do servidor *Web*

Fonte: alterado baseado em Firefox (2021)

Com esses testes foi concluído que o servidor *Web* está conforme a usabilidade.

5.1.5 Teste e análise de usabilidade dos *Workspaces*

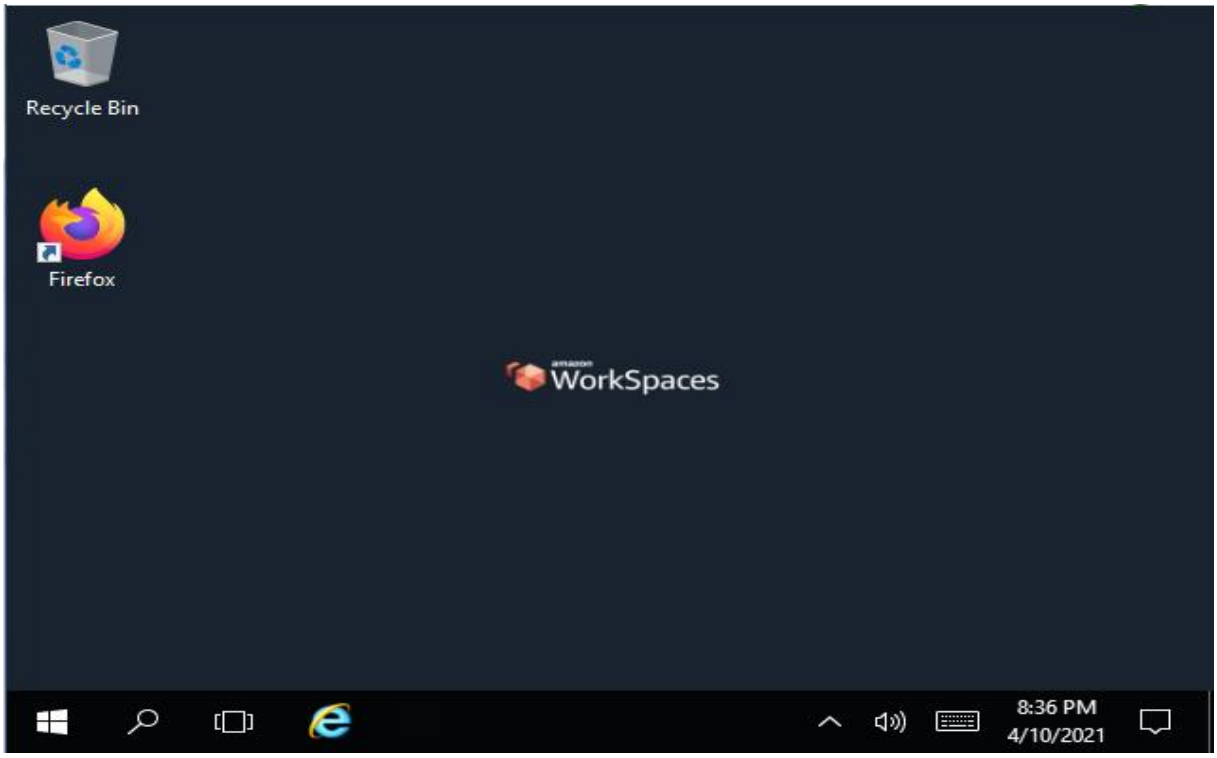
Neste subtópico foram feitos testes de conexão usando o aplicativo *WorkSpaces*, verificando se os usuários cadastrados conseguem autenticar no aplicativo e se esses usuários conseguem acessar os serviços dos servidores FTP e *Web*.

Ao abrir o aplicativo *WorkSpaces*, é solicitado o código de registro. Esse código de registro foi recebido no e-mail após a criação dos usuários. Como mostrado na Figura 103, foi inserido o *user* e *password* do usuário “yannsml” e solicitada autenticação. Na Figura 104 é apresentado a área de trabalho do *Windows* 10, então a autenticação foi bem-sucedida.

Figura 103 – Autenticação *WorkSpaces* usuário yannsml

Fonte: alterado baseado em WorkSpaces (2021)

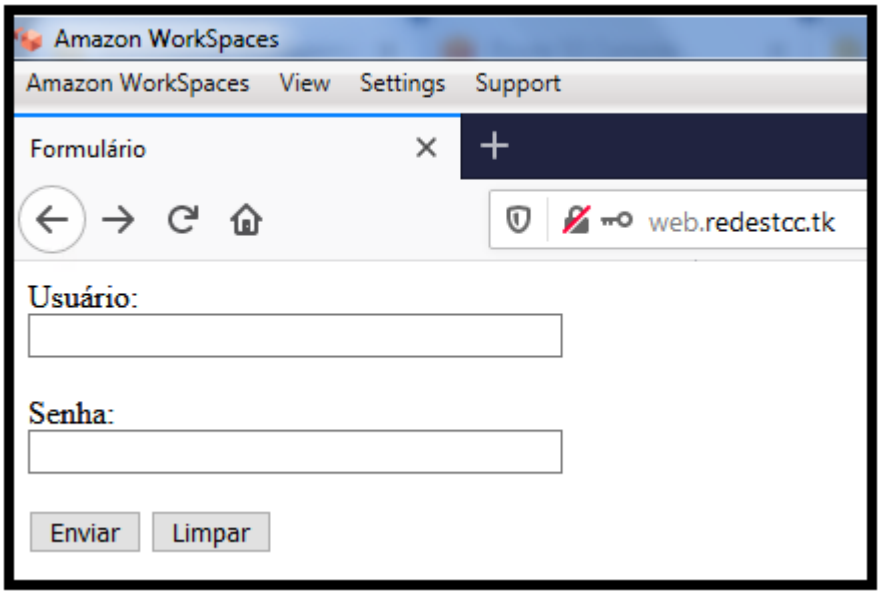
Figura 104 – Área de trabalho *WorkSpaces* usuário yannsml



Fonte: alterado baseado em WorkSpaces (2021)

Para verificar se esse usuário *WorkSpaces* consegue acessar o servidor *Web*, foi aberto o navegador *Firefox* e acessada a URL “web.redetcc.tk”. Como mostrado na Figura 105, a conexão foi bem-sucedida.

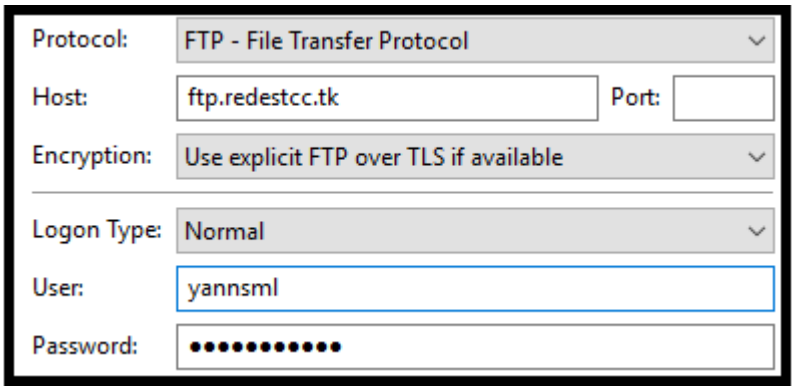
Figura 105 – *WorkSpaces* teste servidor *Web* usuário yannsml



Fonte: alterado baseado em Firefox (2021)

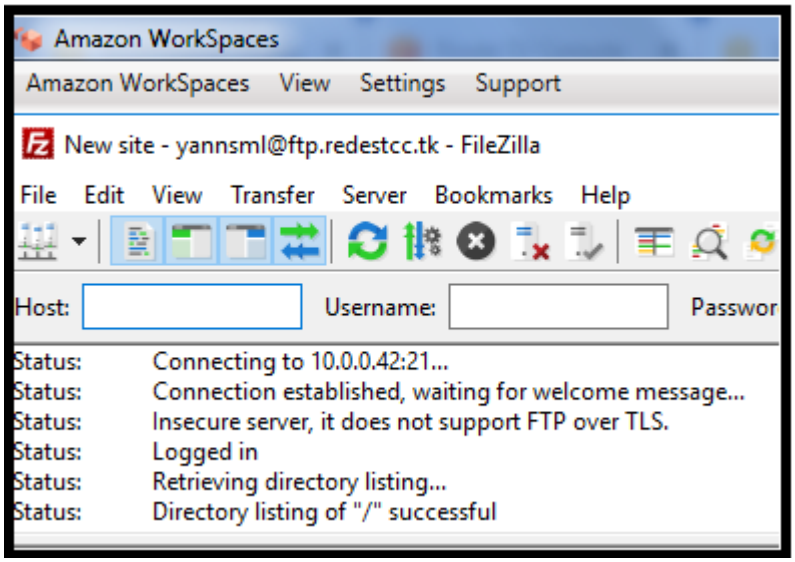
Logado com o usuário “yannsml” no *Workspace*, foi aberto o aplicativo *Filezilla* para testar a conexão com o servidor FTP. Foi inserido as informações de autenticação conforme ilustrado na Figura 106. Como mostrado na Figura 107, a conexão foi bem-sucedida.

Figura 106 – *WorkSpaces* credenciais, servidor FTP usuário yannsml



Fonte: alterado baseado em Filezilla (2021)

Figura 107 – *WorkSpaces* teste servidor FTP usuário yannsml



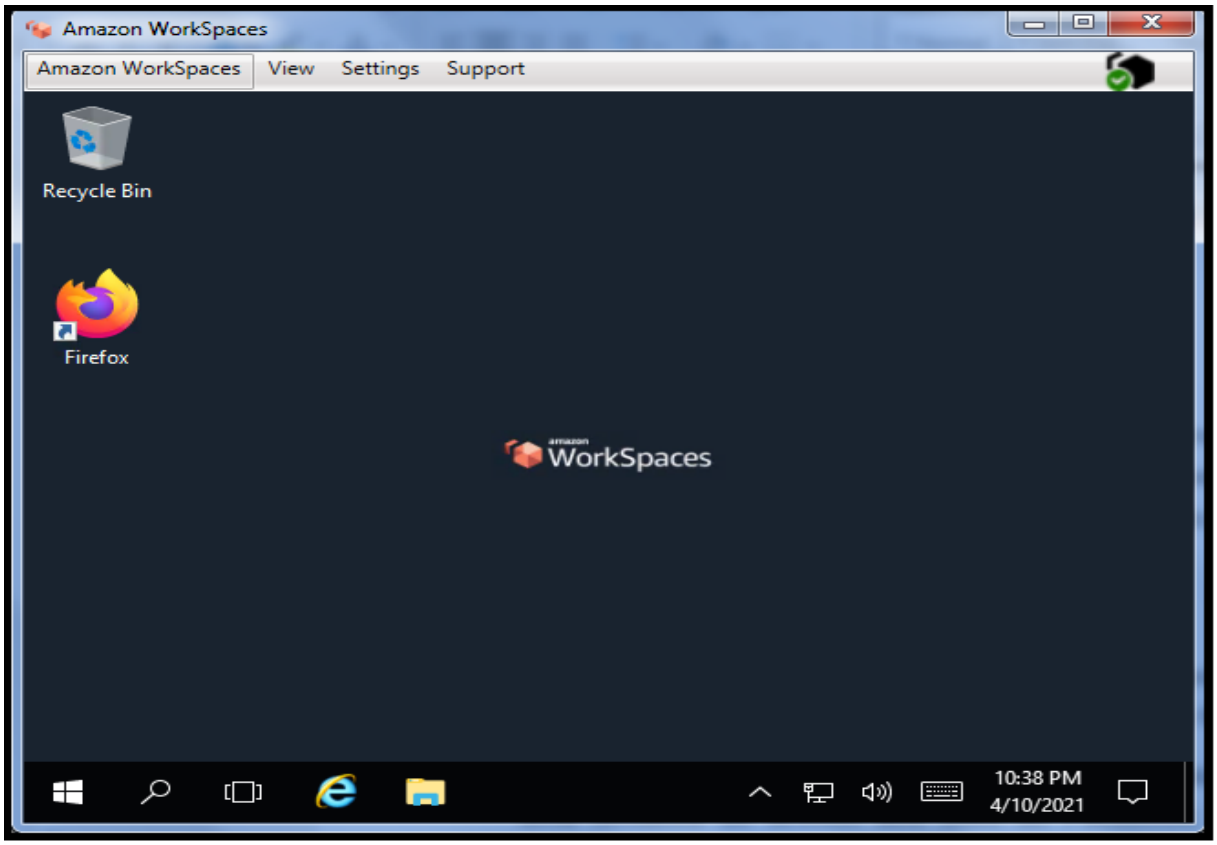
Fonte: alterado baseado em Filezilla (2021)

A seguir, faremos os mesmos testes, porém com o usuário “gabryelsml”. Para isso, foi acessado o aplicativo *WorkSpace* e informado as credenciais informadas na Figura 108. Na Figura 109 é mostrado a área de trabalho do *Windows* 10, então a autenticação foi bem-sucedida.

Figura 108 – Autenticação *WorkSpaces* usuário gabryelsml

Fonte: alterado baseado em WorkSpaces (2021)

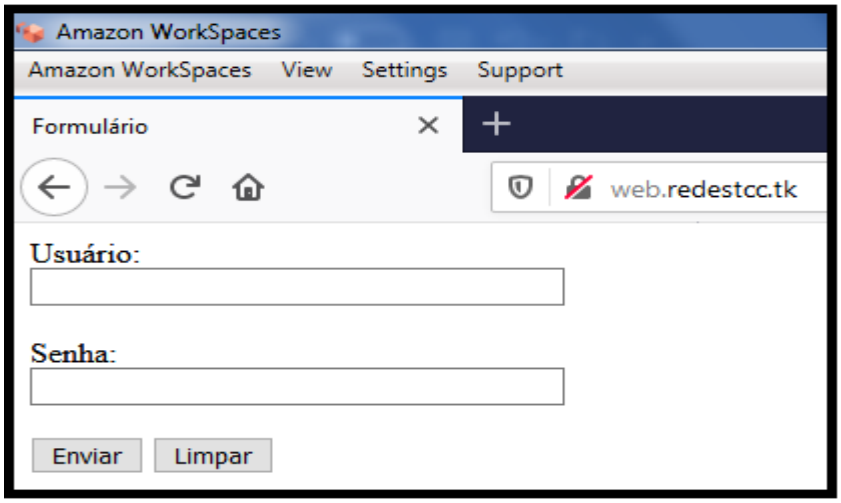
Figura 109 – Área de trabalho *WorkSpaces* usuário gabryelsml



Fonte: alterado baseado em WorkSpaces (2021)

Para verificar a funcionalidade desse usuário *WorkSpaces* em relação ao servidor *Web*, foi aberto o navegador *Firefox* e acessada a URL “web.redetcc.tk”. Como mostrado na Figura 110, a conexão foi bem-sucedida.

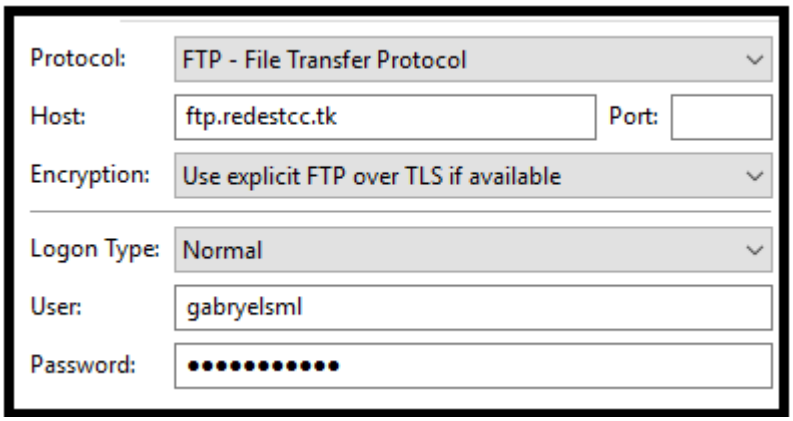
Figura 110 – *WorkSpaces* teste servidor *Web* usuário gabryelsml



Fonte: alterado baseado em Firefox (2021)

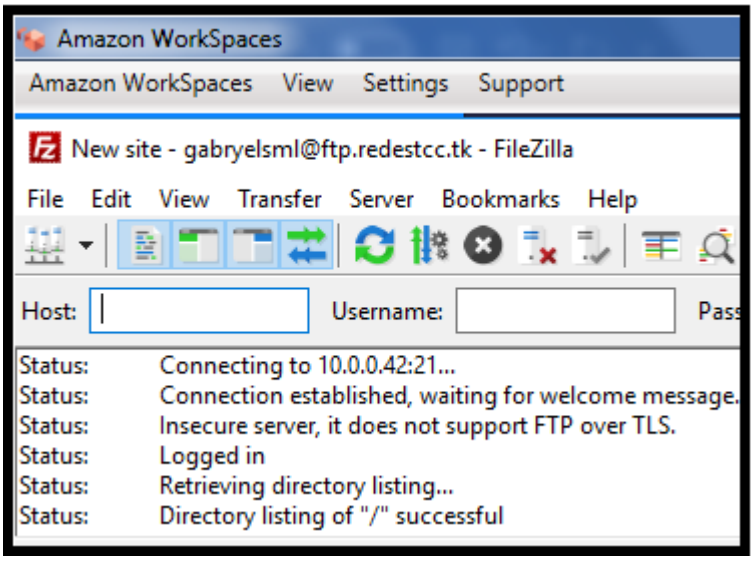
Logado com o usuário “gabryelsml” no *Workspace*, foi aberto o aplicativo *Filezilla* para testar a conexão com o servidor FTP. Foi inserido as informações de autenticação conforme ilustrado na Figura 111. Como mostrado na Figura 112, a conexão foi bem-sucedida.

Figura 111 – *WorkSpaces* credenciais, servidor FTP usuário gabryelsml



Fonte: alterado baseado em Filezilla (2021)

Figura 112 – *WorkSpaces* teste servidor FTP usuário gabryelsml



Fonte: alterado baseado em Filezilla (2021)

Com esses testes foi concluído que o *WorkSpaces* está conforme a usabilidade.

5.2 Teste e análise de confiabilidade

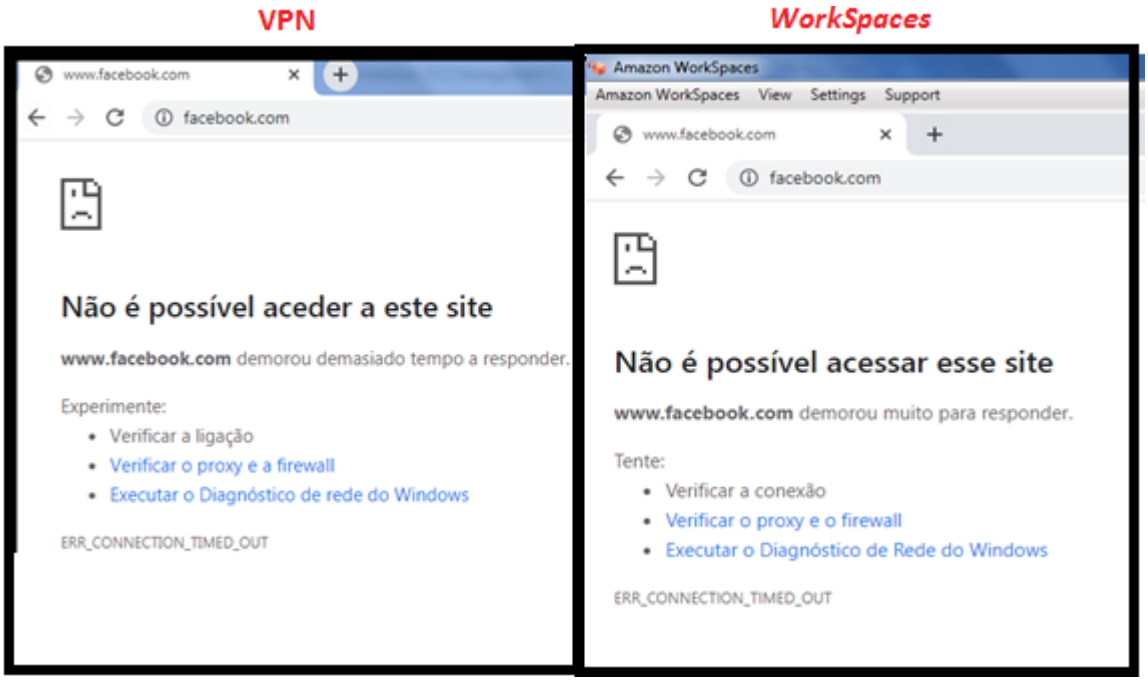
Para teste de confiabilidade, foi testado o envio de pacotes que deverão ser permitidos e pacotes que deverão ser rejeitados pelas ACLs, pelas redes DMZ de cada instância e analisado se está permitindo ou bloqueando os pacotes corretamente.

5.2.1 Teste e análise de confiabilidade da ACL Pública

Os testes realizados para verificar a ACL pública foram testes focados em verificar se essa ACL está bloqueando tráfegos que foram explicitamente proibidos.

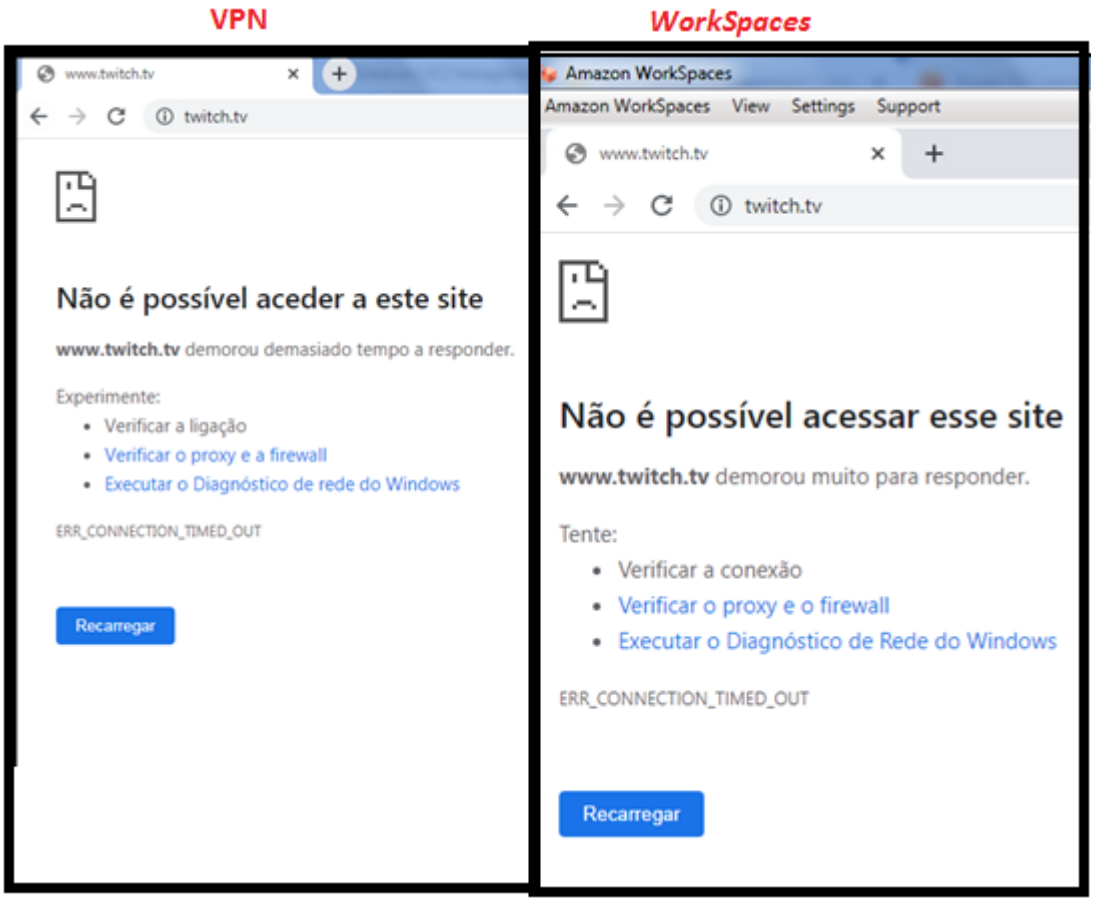
Para testar a ACL pública foi necessário logar no servidor VPN ou no *Workspaces*. Foi realizada uma tentativa de acesso aos sites do *Facebook, Twitch, Amazon Prime Video, Instagram* e Baixaki utilizando o navegador *Google* Chrome. Após esses testes foi realizada tentativa de acesso ao site do *Google*. Conforme ilustrado nas Figuras 113, 114, 115, 116 e 117, não foi estabelecido conexão como esperado, pois, existe regras na ACL pública bloqueando acesso a esses sites, porém, conforme ilustrado na figura 118, foi possível conexão com o *Google*.

Figura 113 – VPN e *Workspaces* conexão *Facebook*



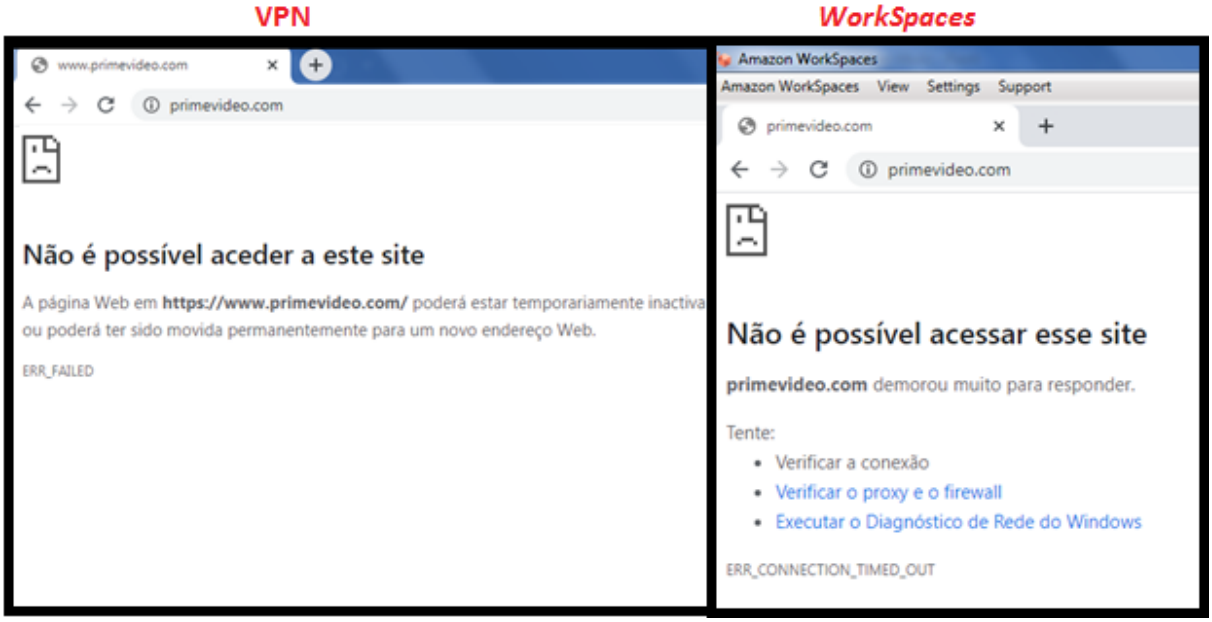
Fonte: alterado baseado em Google Chrome (2021)

Figura 114 – VPN e *Workspaces* conexão *Twitch*



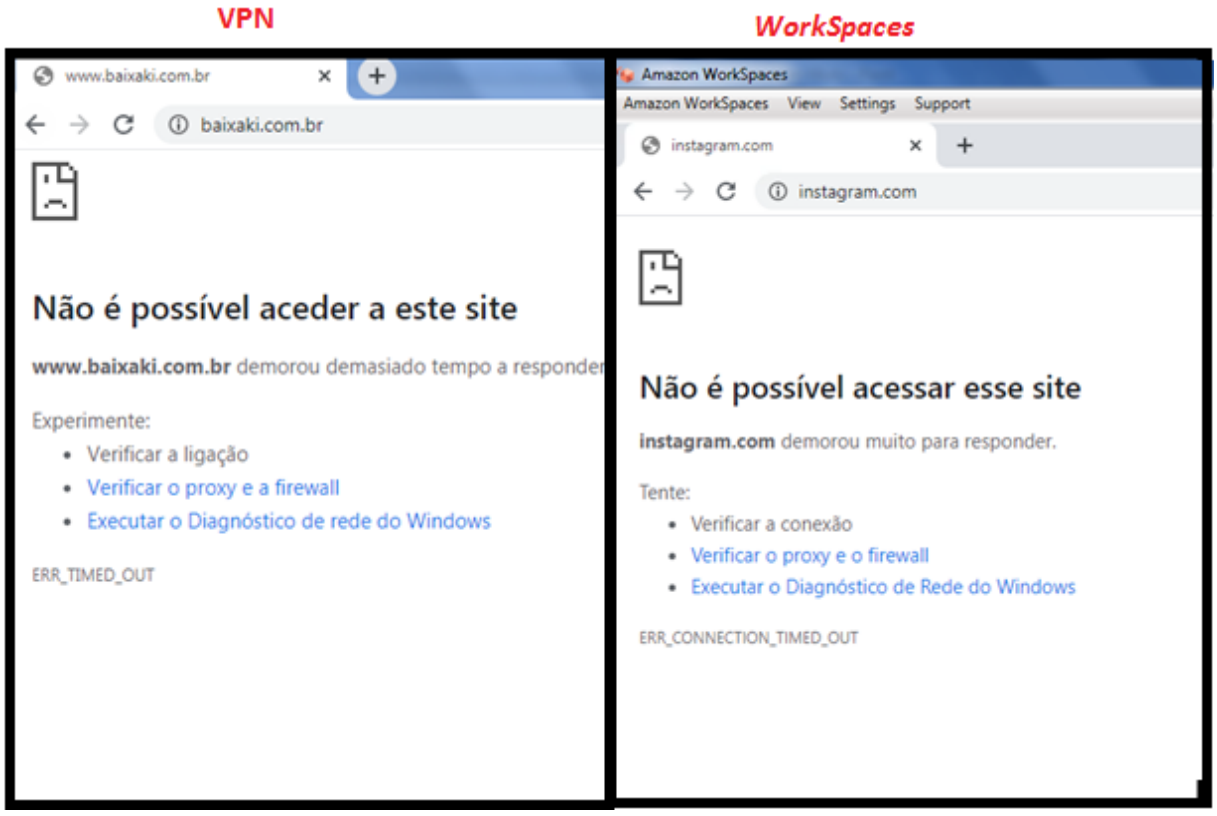
Fonte: alterado baseado em Google Chrome (2021)

Figura 115 – VPN e *WorkSpaces* conexão *Amazon Prime Video*



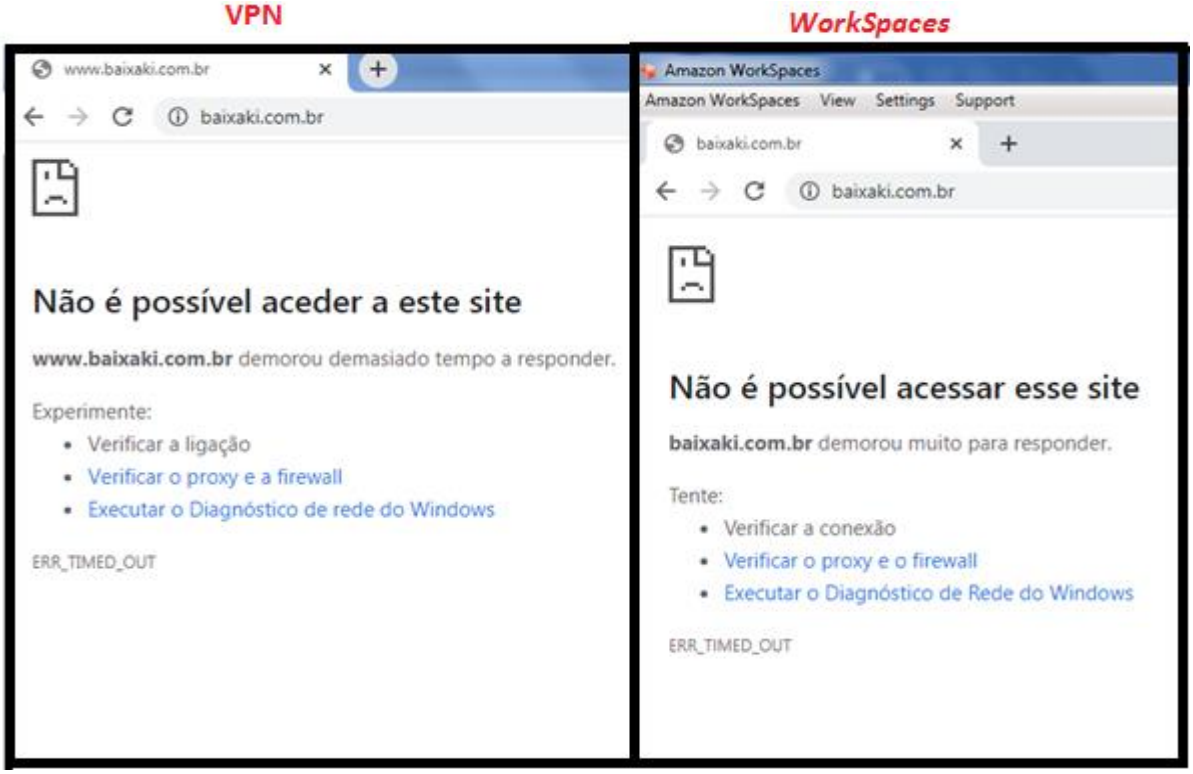
Fonte: alterado baseado em Google Chrome (2021)

Figura 116 – VPN e *WorkSpaces* conexão *Instagram*



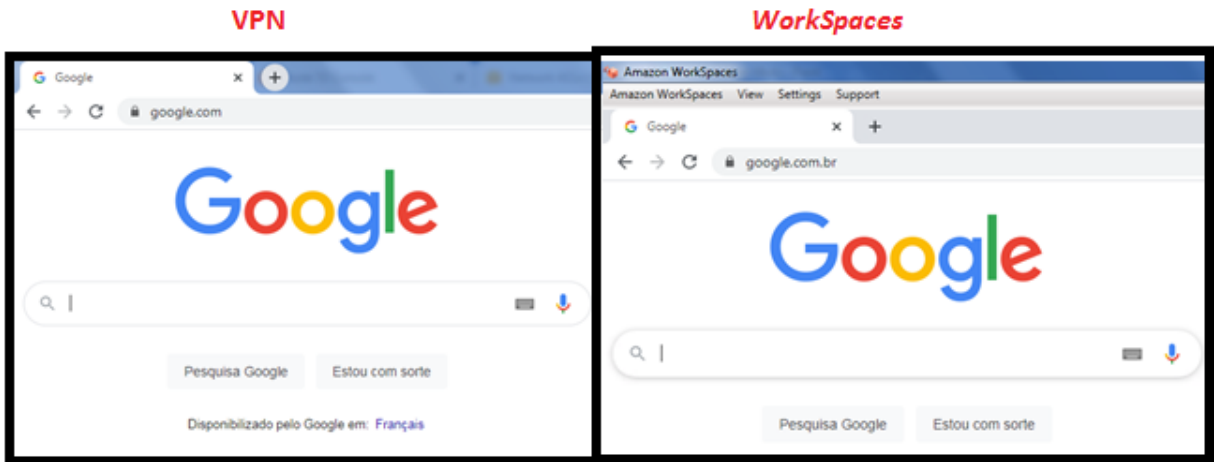
Fonte: alterado baseado em Google Chrome (2021)

Figura 117 – VPN e *Workspaces* conexão Baixaki



Fonte: alterado baseado em Google Chrome (2021)

Figura 118 – VPN e *WorkSpaces* conexão *Google*



Fonte: alterado baseado em Google Chrome (2021)

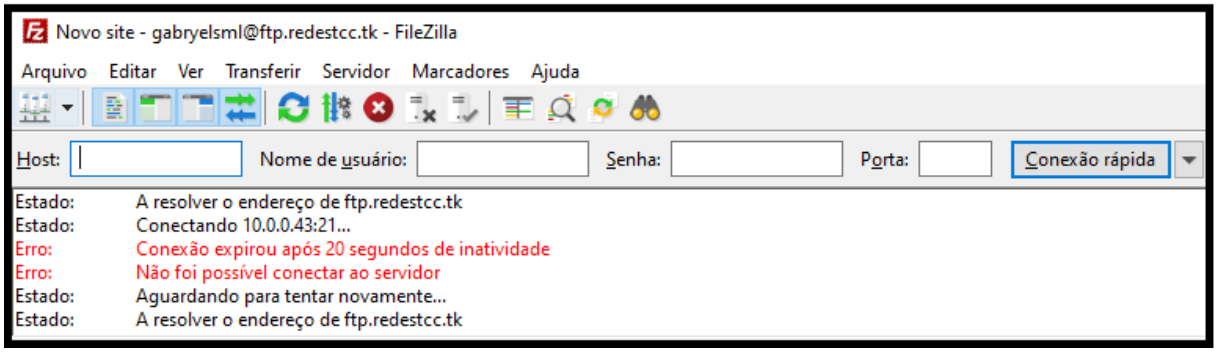
Portanto, foi concluído que a ACL pública está conforme a confiabilidade.

5.2.2 Teste e análise de confiabilidade da ACL Privada

Os testes realizados para verificar a ACL privada foram focados em verificar se a sub-rede privada somente é acessada pelas máquinas pertencentes a rede VPC.

Para testar a ACL privada, foi realizado teste para tentar conexão com o servidor FTP via aplicativo *FileZilla* sem estar logado na VPN. Foi utilizado o usuário “gabryelsml” para logar no servidor FTP. Como mostrado na Figura 119, a conexão não foi bem-sucedida

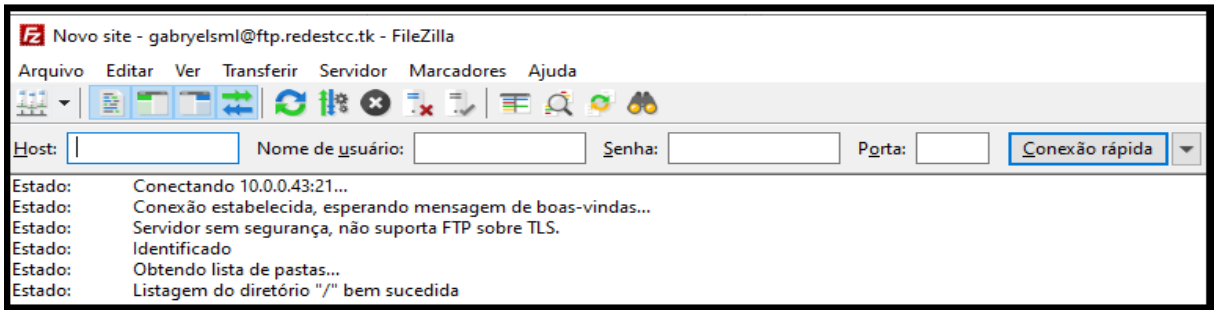
Figura 119 – Conexão FTP *FileZilla* sem VPN



Fonte: alterado baseado em FileZilla (2020)

A seguir, foi logado no servidor VPN e foi realizado teste tentando logar com o usuário informado anteriormente. Como mostrado na Figura 120, a conexão foi bem-sucedida e usuário logado no servidor FTP.

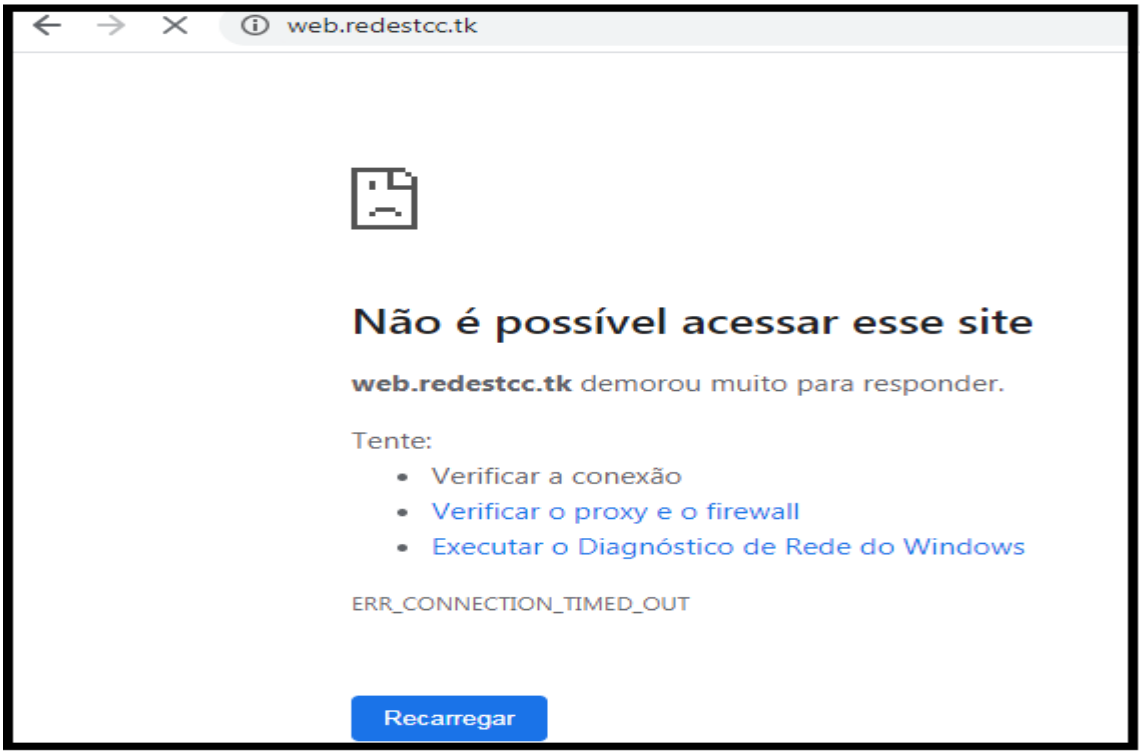
Figura 120 – Conexão FTP *FileZilla* com VPN



Fonte: alterado baseado em FileZilla (2020)

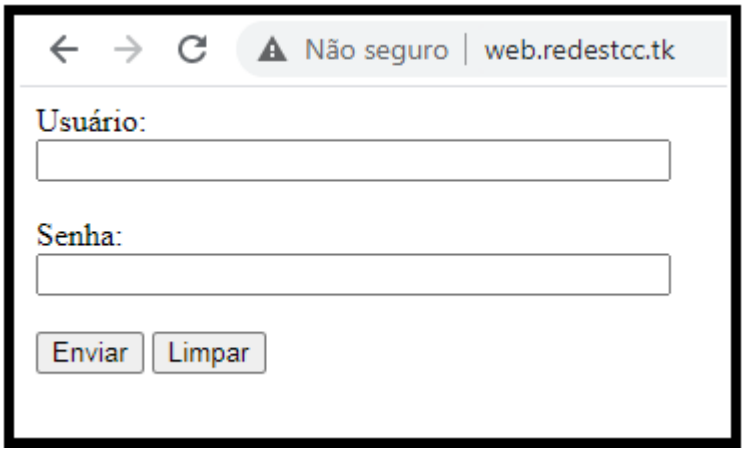
Para testar a ACL privada em relação ao servidor *Web*, foi realizado teste tentando acessar o site “web.redestcc.tk” sem logar no servidor VPN e no servidor *Web Proxy,* e conforme mostrado na Figura 121, não foi possível conexão. Porém após logar no servidor VPN, foi possível comunicar com o site “web.redestcc.tk” como mostrado na Figura 122.

Figura 121 – Conexão *Web* sem VPN



Fonte: alterado baseado em Google Chrome (2021)

Figura 122 – Conexão *Web* com VPN



Fonte: alterado baseado em Google Chrome (2021)

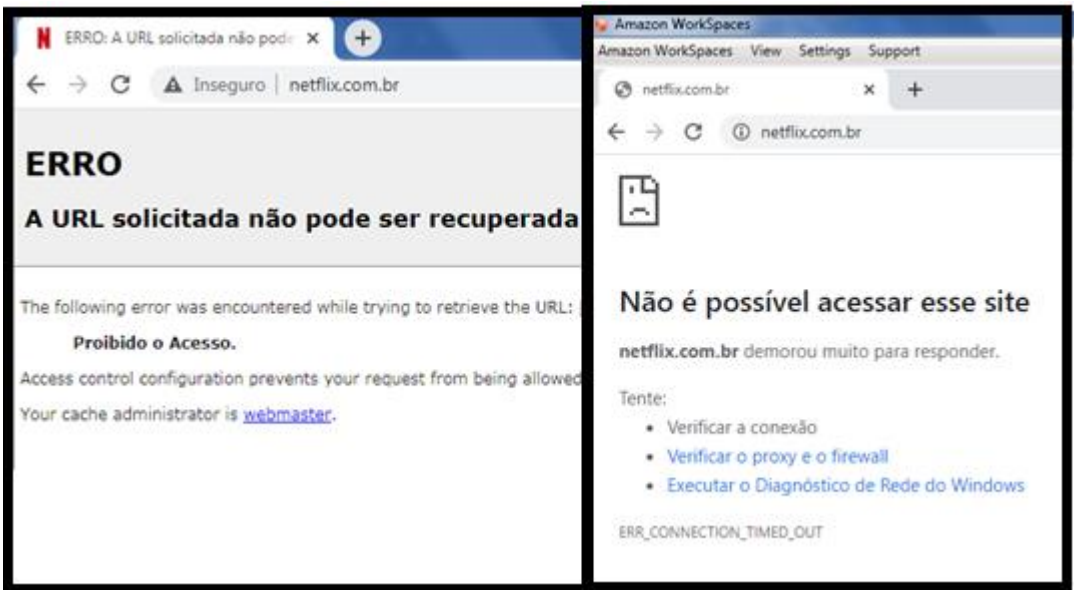
Portanto, a ACL privada está conforme a confiabilidade.

5.2.3 Teste e análise de confiabilidade do Servidor *Web Proxy*

Os testes realizados para verificar a confiabilidade do Servidor *Web Proxy* foram testes focados em verificar se o servidor está bloqueando acesso a sites que foram proibidos.

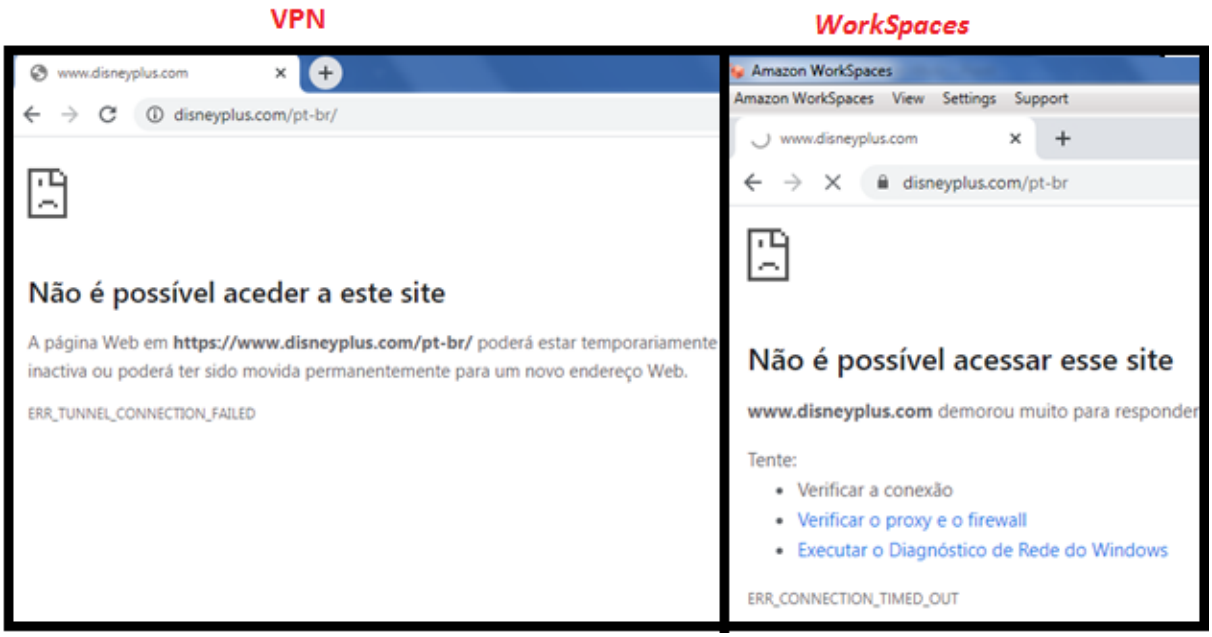
Para testar o *Web Proxy* está bloqueando corretamente, foi realizada tentativa de acessar os sites do *Netflix*, *Twitter* e *Disney* logando no servidor VPN ou no *WorkSpaces*. Após isso, foi realizada tentativa de entrar no *google*. Como ilustrado nas Figuras 123, 124 e 125, não foi possível acesso aos sites do *Netflix, Twitter* e *Disney*. Porém conforme ilustrado na Figura 126, foi possível acesso ao site do *Google*.

Figura 123 – *Web Proxy* usando VPN e *WorkSpaces Netflix*



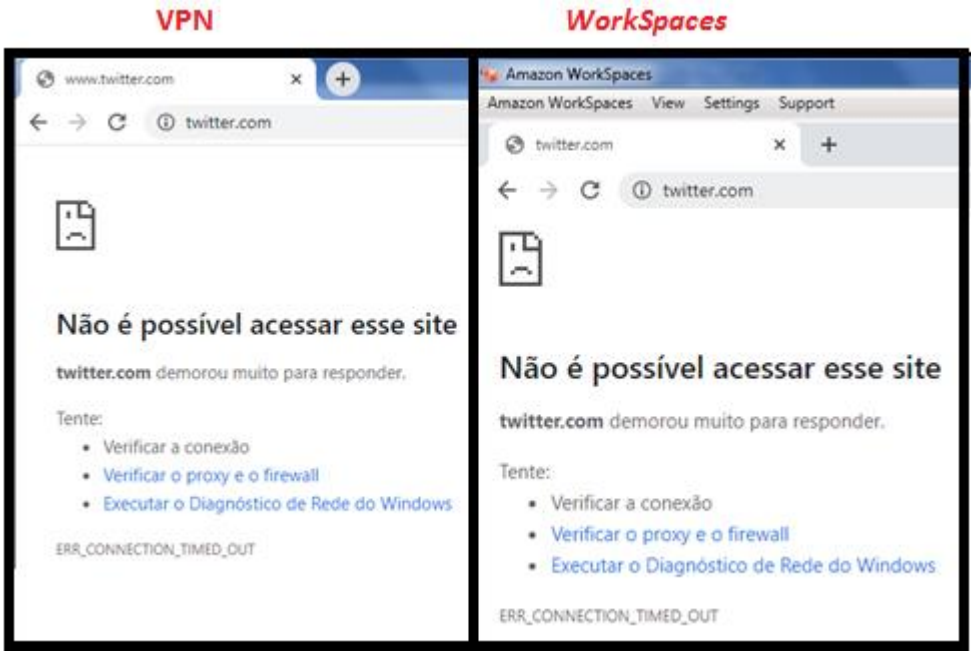
Fonte: alterado baseado em Google Chrome (2021)

Figura 124 – *Web Proxy* usando VPN e *WorkSpaces Disney*



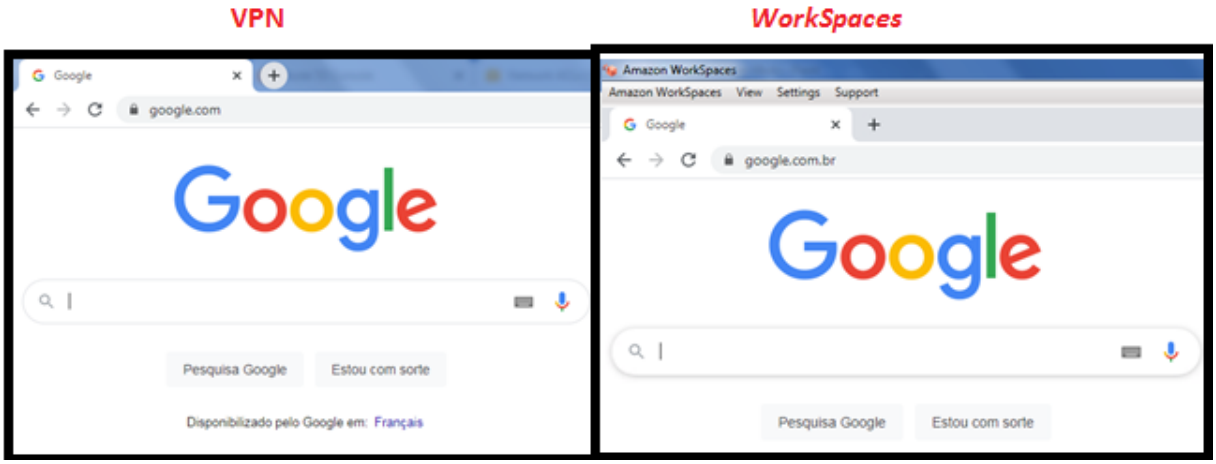
Fonte: alterado baseado em Google Chrome (2021)

Figura 125 – *Web Proxy* usando VPN e *WorkSpaces Twitter*



Fonte: alterado baseado em Google Chrome (2021)

Figura 126 – *Web Proxy* usando VPN e *Workspaces Google*



Fonte: alterado baseado em Google Chrome (2021)

Portanto, após todos esses testes concluiu-se que o servidor *Web Proxy* está conforme a confiabilidade.

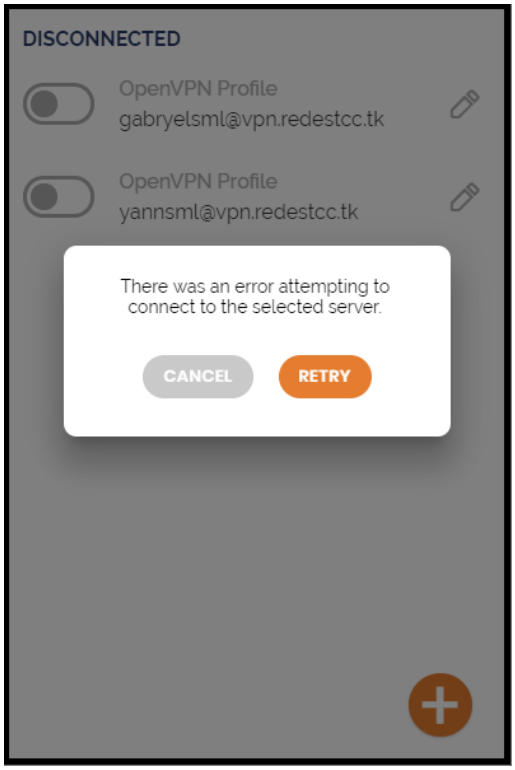
5.2.4 Teste e análise de confiabilidade da Rede DMZ *Security Group*

Os testes realizados para verificar as redes DMZ das instâncias foram focadas em testar conexões que não foram explicitamente permitidas nas regras da *Security Group* e analisado se está permitindo ou bloqueando os pacotes corretamente.

5.2.4.1 Teste e análise de confiabilidade da Rede DMZ VPN

Para testar a rede DMZ do servidor VPN, foram retiradas as regras que permitiam a entrada de tráfego nas portas UDP 1194, TCP 943 e 443 na *Security Group* e foi realizada uma tentativa de logar no *OpenVPN* com o usuário “gabryelsml”. Conforme ilustrado na Figura 127, não foi possível realizar *login*, portanto a rede DMZ está bloqueando o tráfego corretamente.

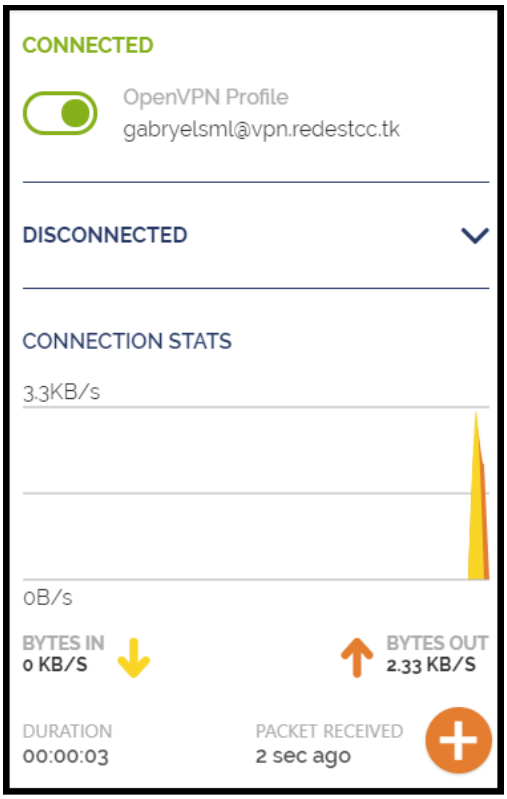
Figura 127 – *Login* servidor VPN sem regra na DMZ



Fonte: alterado baseado em OpenVPN (2020)

A seguir, foram inseridas as regras que permitiam a entrada de tráfego nas UDP 1194, TCP 943 e 443 na *Security Group* e foi realizado uma tentativa de logar no *OpenVPN* com o usuário “gabryelsml”. Conforme ilustrado na Figura 128, o usuário foi logado com sucesso.

Figura 128 – *Login* servidor VPN com regra na DMZ



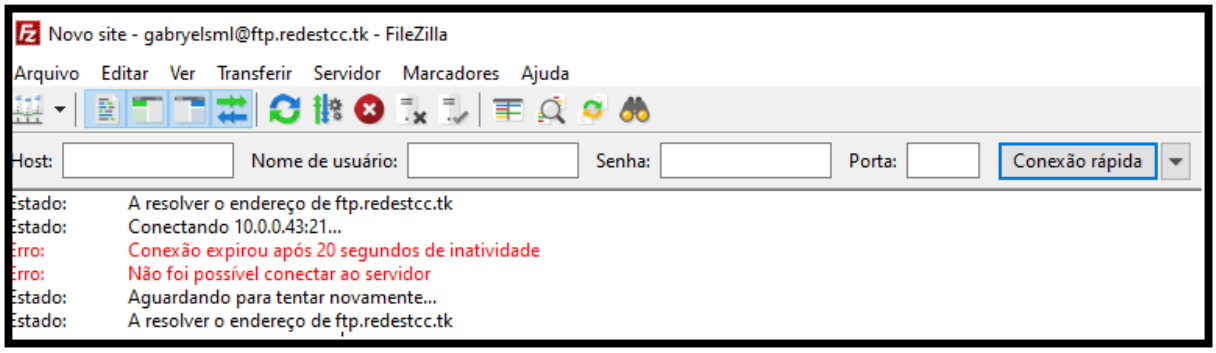
Fonte: alterado baseado em OpenVPN (2020)

Portanto a rede DMZ do servidor VPN está permitindo e bloqueando tráfego corretamente.

5.2.4.2 Teste e análise de confiabilidade da Rede DMZ FTP

Para testar a rede DMZ do servidor FTP, foi realizado *login* no servidor VPN com o usuário “gabryelsml” e foram retiradas as regras que permitiam a entrada de tráfego nas portas 1025 a 1029 e da porta 20 a 21 da *Security Group*. A seguir, foi realizada uma tentativa de logar no servidor FTP via *FileZilla*. Como ilustrado na Figura 129, a conexão não foi bem-sucedida, portanto, a rede DMZ está bloqueando o tráfego corretamente.

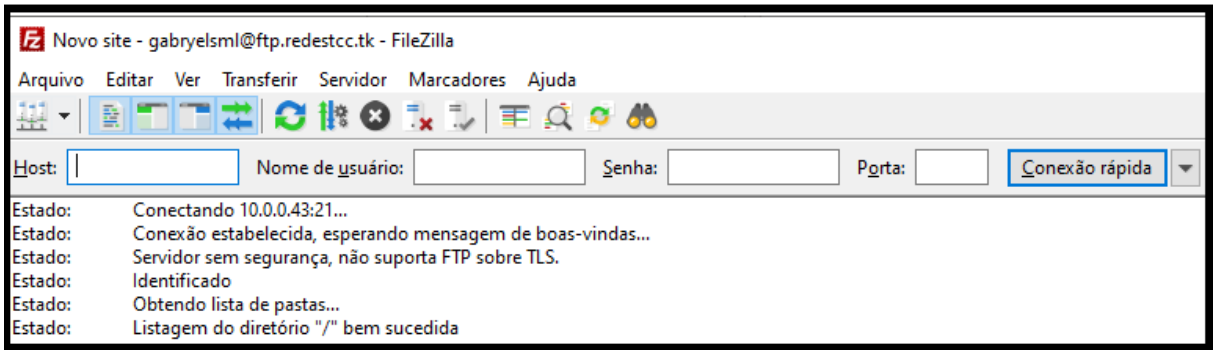
Figura 129 – Logar servidor FTP sem regras DMZ



Fonte: alterado baseado em FileZilla (2020)

Foram inseridas as regras que permitiam a entrada de tráfego nas portas 1025 a 1029 e da porta 20 a 21 da *Security Group* e foi realizada uma tentativa de logar no servidor FTP via *FileZilla*. Como ilustrado na Figura 130, o *login* foi bem-sucedido.

Figura 130 - Logar servidor FTP com regras DMZ



Fonte: alterado baseado em FileZilla (2020)

Portanto, a rede DMZ do servidor FTP está permitindo e bloqueando tráfego corretamente.

5.2.4.3 Teste e análise de confiabilidade da Rede DMZ *Web Proxy*

Para realizar teste na Rede DMZ do servidor *Web Proxy*, foi realizado *login* na VPN com o usuário “yann.sml”. Foi retirada a regra de entrada de dados que permite a passagem de dados TCP pela porta 3128 da *Security Group*. Foi informado o nome “webproxy.redestcc.tk” que roteia para o IP público do servidor Web Proxy e a porta 3128 e realizou-se um teste abrindo o navegador *Google Chrome*.

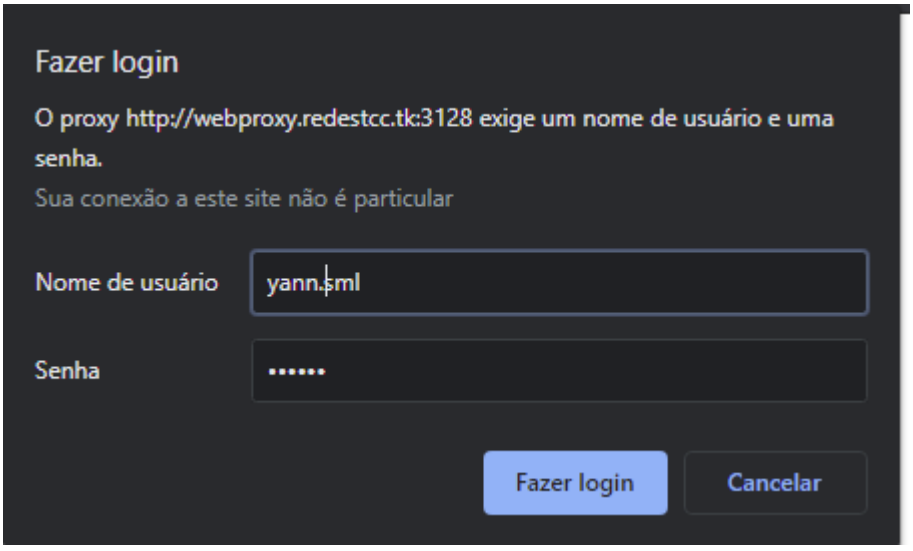
Conforme mostrado na Figura 131, não houve solicitação para informar *login* e senha, e não foi possível comunicação com a *Internet*.

Figura 131 – Sem regras na DMZ Web Proxy

Fonte: alterado baseado em Google Chrome (2020)

A seguir, foi inserido a regra que permite passagem de dados TCP na porta 3128. Ainda logado na VPN e com o *Web Proxy* configurado no computador, foi realizado teste abrindo o navegador *Google Chrome*. Conforme Figura 132, foi solicitado *login* e senha para autenticar no servidor *Web Proxy*.

Figura 132 – Com regras na DMZ Web Proxy autenticação



Fonte: alterado baseado em Google Chrome (2020)

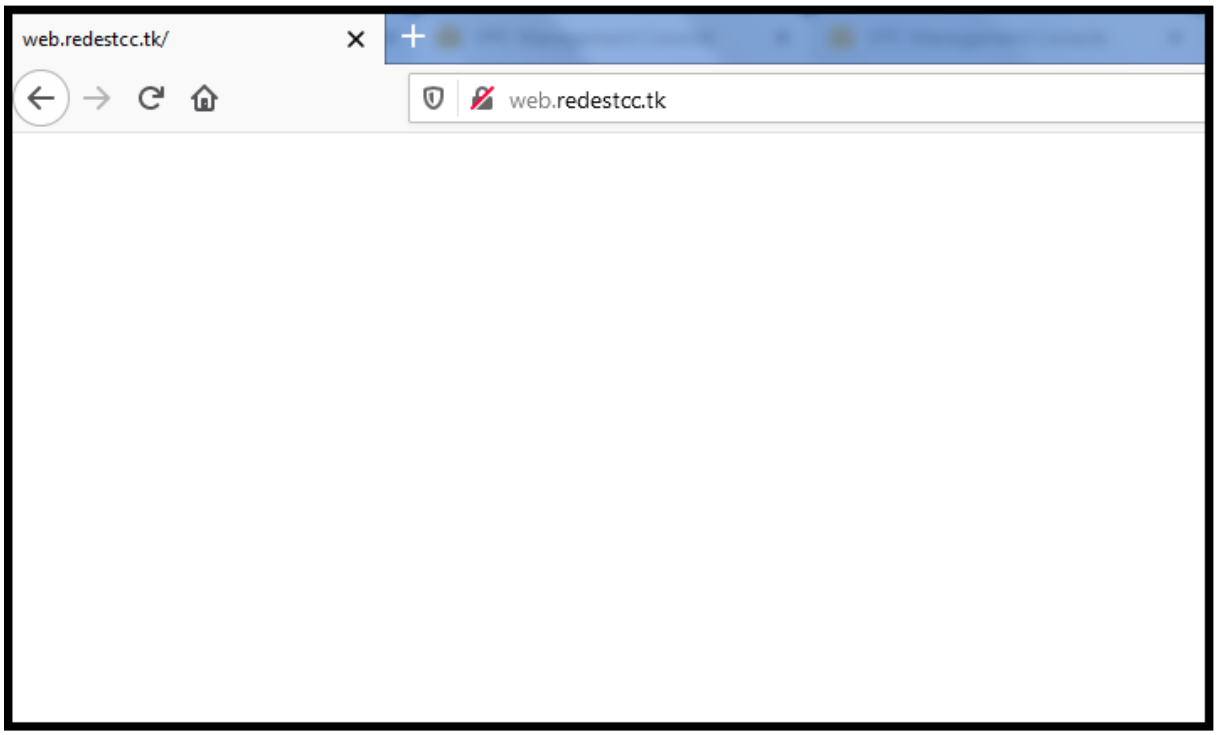
A seguir, foi inserido o *login* e a senha, e solicitado a autenticação. Ao fim, foi realizada comunicação com a *Internet*. Portanto a rede DMZ do servidor *Web Proxy* está permitindo e bloqueando tráfego corretamente.

5.2.4.4 Teste e análise de confiabilidade da Rede DMZ *Web*

Para teste da Rede DMZ do servidor *Web*¸ foi feito *login* no servidor VPN e feita autenticação no servidor *Web Proxy*. A seguir, foi retirada a regra de entrada de dados que permite a passagem de dados TCP pela porta 80 (Conexão HTTP) da *Security Group*.

Foi aberto o navegador *Firefox,* inserido a URL “web.redestcc.tk” e foi solicitada conexão. Como mostrado na Figura 133, não foi possível conexão, porém não retornou nenhum tipo de erro.

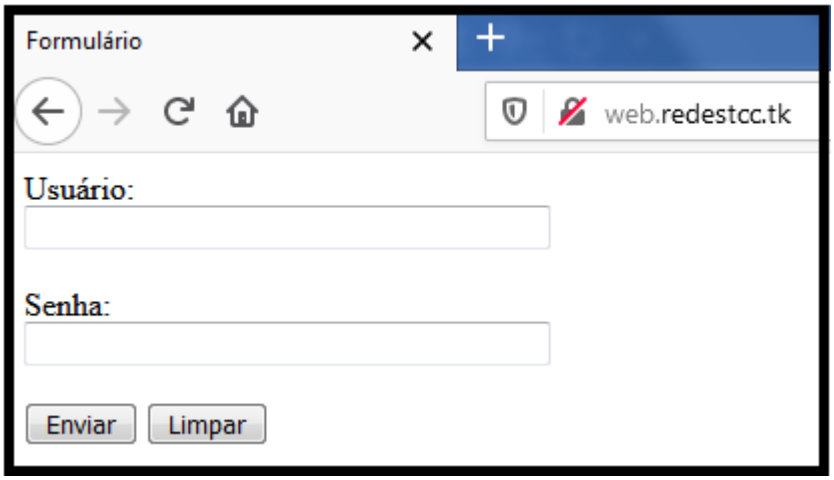
Figura 133 – Sem regras na DMZ servidor *Web*



Fonte: alterado baseado em Firefox (2021)

A seguir, ainda logado no servidor VPN e *Web Proxy,* foi incluído novamente a regra de entrada de dados que permite a passagem de dados TCP pela porta 80 da *Security Group,* foi aberto o navegador *Firefox,* inserido a URL “web.redestcc.tk” e foi solicitada conexão. Como apresentado na Figura 134, a conexão foi bem-sucedida.

Figura 134 – Com regras na DMZ servidor *Web*



Fonte: alterado baseado em Firefox (2021)

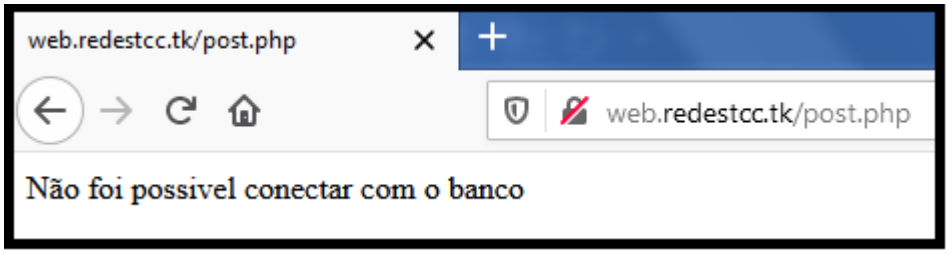
Portanto a rede DMZ do servidor *Web* está permitindo e bloqueando tráfego corretamente.

5.2.4.5 Teste e análise de confiabilidade da Rede DMZ RDS

Para teste da Rede DMZ do banco de dados¸ foi feito *login* no servidor VPN e feita autenticação no servidor *Web Proxy*. A seguir, foi retirada a regra de entrada de dados que permite a passagem de dados TCP pela porta 3306 (Conexão MySQL) da *Security Group*.

Foi aberto o navegador *Firefox,* inserido a URL “web.redestcc.tk” e foi solicitada conexão. A seguir, foi informado o usuário e senha do usuário FTP “yannsml” e enviado a requisição. Como ilustrado na Figura 135, não foi possível conexão com o banco de dados.

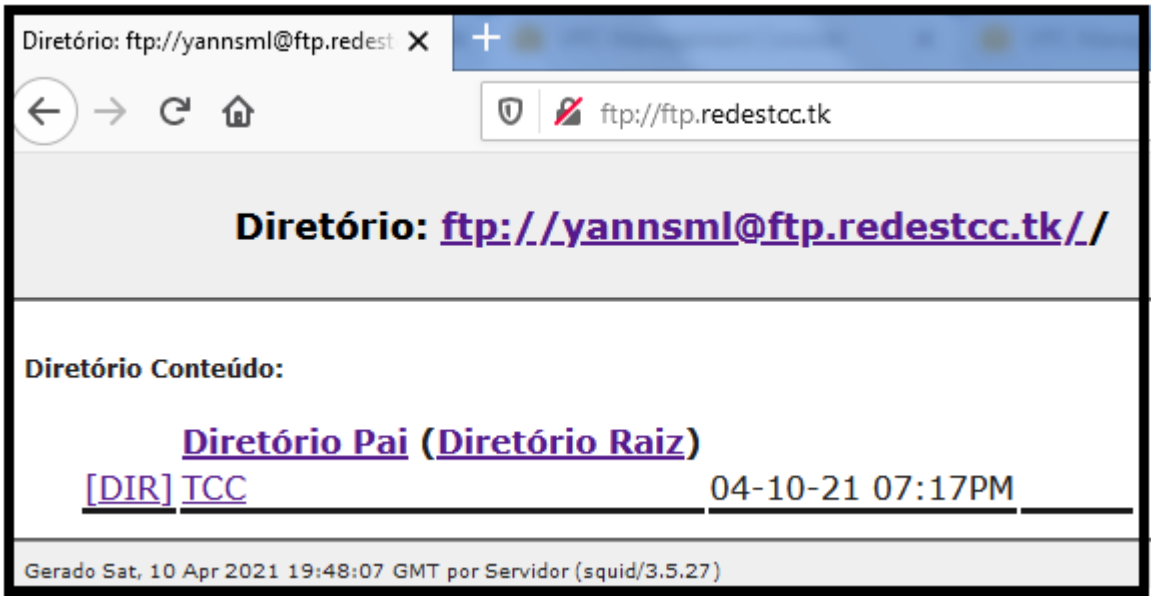
Figura 135 – Sem regras na DMZ banco de dados



Fonte: alterado baseado em Firefox (2021)

Ainda logado no servidor VPN e *Web Proxy*, foi inserido novamente a regra de entrada que permite a passagem de dados TCP pela porta 3306 da *Security Group*, foi solicitado o acesso a página “web.redestcc.tk” utilizando o navegador *Firefox,* foi informado o usuário “yannsml” e sua senha e foi solicitado a conexão. Como mostrado na Figura 136, a página foi redirecionada para a página do servidor FTP, portanto, houve conexão com o banco de dados.

Figura 136 – Com regras na DMZ banco de dados



Fonte: alterado baseado em Firefox (2021)

Devido a esses testes, podemos afirmar que a rede DMZ do banco de dados está permitindo e bloqueando tráfego corretamente.

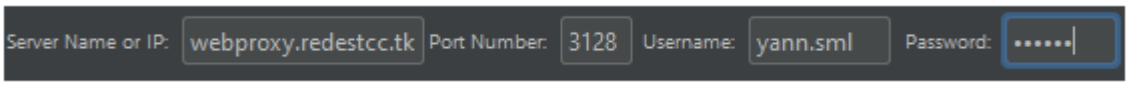
5.2.5 Teste de carga

Esse teste de carga tem como objetivo verificar se a ACL pública e servidor *Web Proxy* conseguem bloquear sites (que devem ser bloqueados de acordo com suas regras) mesmo com uma grande quantidade de solicitações.

Esse teste foi realizado após *login* na VPN e inserido o *Proxy*. Foram realizadas simulações utilizando o aplicativo JMeter, com 500 usuários realizando solicitações HTTP para os sites do *Facebook, Twitch, Twitter,* Baixaki*, Amazon Prime Video, Netflix, Instagram* e *Disney*, simultaneamente, por 10 vezes. Assim, foram realizadas 5.000 solicitações para cada um dos sites acima, totalizando 40.000 solicitações. No aplicativo foi inserido o nome “webproxy.redestcc.tk” que roteia para o IP público do

servidor Web Proxy, sua porta, o *login* do usuário e sua senha, conforme mostrado na Figura 137.

Figura 137 – *Proxy JMeter*



Fonte: alterado baseado em Google Chrome (2020)

Conforme ilustrado na Figura 138, houve 100% de erro em todas as 40.000 solicitações de acesso aos sites informados. Esse teste levou cerca de 25 minutos para executar.

Figura 138 – Resultado teste de carga

Label	# Samp...	Average	Median	90% Line	95% Li...	99% Line	Min	Maxim...	Error %
facebook	5000	21160	21070	21241	21590	22912	20987	23244	100.00%
twitter	5000	117	100	165	206	290	75	1604	100.00%
disney	5000	114	97	167	207	269	76	1254	100.00%
twitch	5000	21096	21061	21139	21311	22059	20985	22289	100.00%
Amazon prime	5000	1096	1055	1181	1428	1670	996	2320	100.00%
netflix	5000	99	93	122	133	152	76	1143	100.00%
baixaki	5000	374362	378539	378672	378706	378782	357305	379010	100.00%
instagram	5000	21029	21028	21050	21062	21095	20986	21142	100.00%
TOTAL	40000	54884	2320	357830	378567	378682	75	379010	100.00%

Fonte: alterado baseado em Jmeter (2020)

Portanto, a ACL pública e o servidor *Web Proxy* suportam várias solicitações simultaneamente sem perder a confiabilidade.

5.3 Teste e análise de desempenho

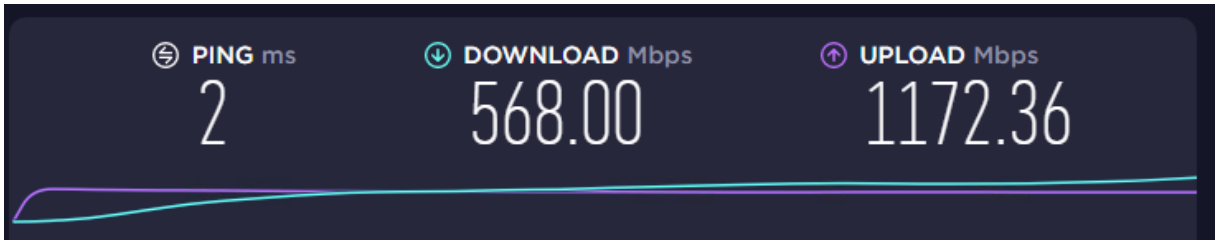
Para verificar o desempenho, foi testada a velocidade de *download* e *upload* de um cliente deslogado, logado na VPN e no *Workspaces*, e feito a comparação entre eles. Foi testada a taxa de *download* e *upload* de arquivos no servidor FTP, e para análise foi realizada a comparação entre os clientes *WorkSpaces* e VPN.

5.3.1 *Teste e análise de Download e Upload Clientes VPN e Workspaces*

Foi realizado *login* usando o usuário “yannsml” no aplicativo *WorkSpace*, foi aberto o navegador Google *Chrome* e foi acessado o site da *SpeedTest*. Como

mostrado na Figura 139, o cliente *WorkSpace* possui 568 Mbps de *download* e 1172,36 Mbps de *upload*.

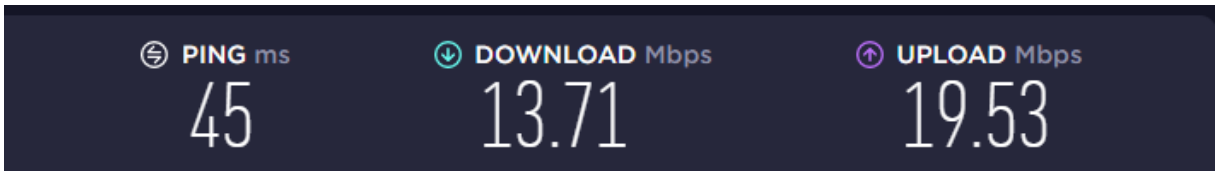
Figura 139 – Velocidade de cliente *WorkSpaces*



Fonte: alterado baseado em SpeedTest (2021)

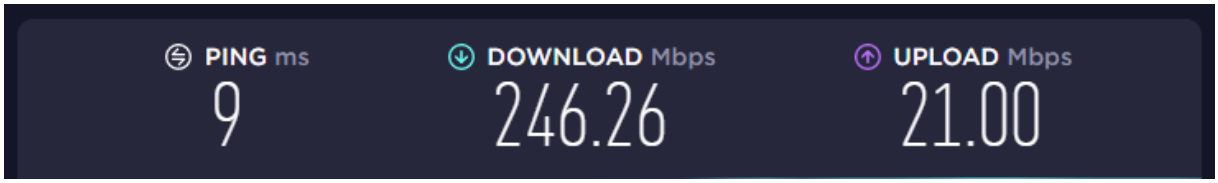
Foi realizado *login* utilizando o usuário “yannsml” no servidor VPN, foi acessado *Google Chrome* e foi acessado o site da *SpeedTest*. Conforme mostrado na Figura 140, o cliente VPN possui 13,48 Mbps de *download* e 19,53 Mbps de *upload.* Ao deslogar no servidor VPN, foi realizado o mesmo teste e o resultado foi de 246,26 Mbps de *download* e 21 Mbps de *upload* conforme ilustrado na Figura 141.

Figura 140 – Velocidade de cliente VPN



Fonte: alterado baseado em SpeedTest (2021)

Figura 141 – Velocidade do cliente deslogado



Fonte: alterado baseado em SpeedTest (2021)

Foi comparando o cliente logado no aplicativo *WorkSpaces* com o cliente logado no aplicativo *OpenVPN*. Foi observado que houve um ganho de 4042% de *download* e 5902% de *upload* usando o *WorkSpaces*.

Foi comparando o cliente logado no aplicativo *WorkSpaces* com o cliente deslogado. Observou-se que houve um ganho de 130% de *download* e 5482% de *upload* quando o cliente está logado no *WorkSpaces*.

Foi comparando o cliente logado no aplicativo *OpenVPN* com o cliente deslogado. Percebeu-se que houve um ganho de 1696% de *download* e 7% de *upload* quando o cliente está deslogado.

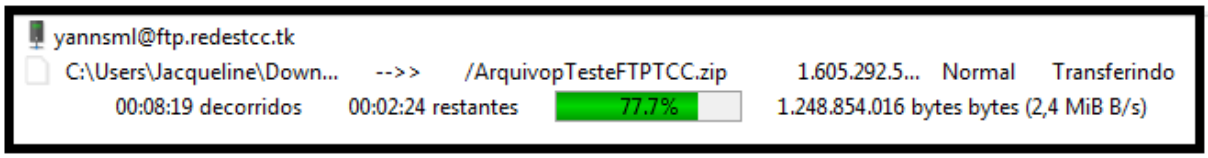
Portanto, a utilização dos *WorkSpaces* proporciona desempenho além de ser consideravelmente mais veloz em comparação com a VPN e com clientes que estejam deslogados do servidor VPN e do aplicativo *WorkSpaces*. Também foi verificado que ao logar no servidor VPN, houve perda de desempenho.

5.3.2 Teste e análise de *Download* e *Upload* de arquivos FTP

Nestes testes foi utilizado um arquivo chamado “ArquivopTesteFTPTCC.zip” com tamanho de 1.605.292.516 *bytes* ou 1,605 GB para verificar o tempo de download e *upload* do arquivo usando a VPN e o *Workspaces*.

Foi realizado *login* no *OpenVPN* utilizando o usuário “yannsml”, foi realizado *login* no *FileZilla* com esse mesmo usuário e foi solicitado envio do arquivo para o servidor FTP. Conforme apresentado na Figura 142, a taxa de *upload* teve em média 2,4 Mbps. Este *upload* demorou 10 minutos e 43 segundos (643 segundos).

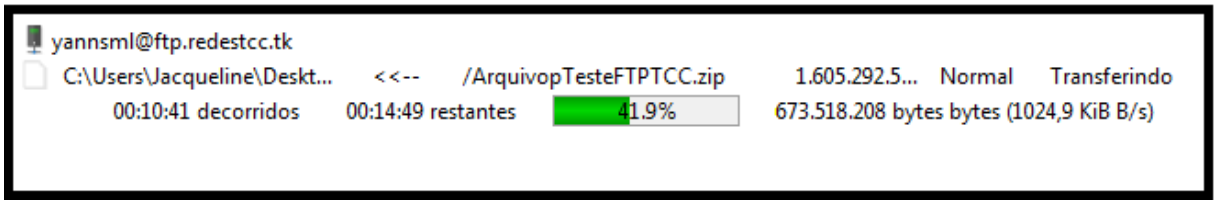
Figura 142 – Taxa de *Upload FileZilla* VPN



Fonte: alterado baseado em FileZilla (2021)

Ainda logado no servidor VPN e FTP, foi solicitado o *download* do arquivo para a máquina do cliente VPN. Conforme ilustrado na Figura 143, a taxa de *download* teve em média 1 Mbps. Este *download* demorou 25 minutos e 40 segundos (1540 segundos).

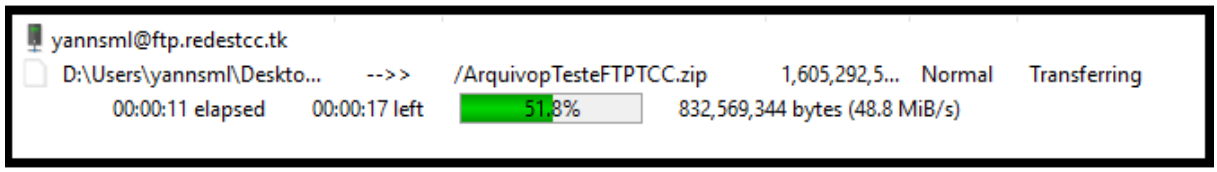
Figura 143 – Taxa de *Download FileZilla* VPN



Fonte: alterado baseado em FileZilla (2021)

Foi aberto o aplicativo *WorkSpaces* e foi feito *login* com o usuário “yannsml”. Para logar no servidor FTP, foi utilizado esse mesmo usuário. Foi solicitado o envio do arquivo para o servidor FTP. Como mostrado na Figura 144, a taxa de *upload* teve em média 48,8 Mbps. Este *upload* demorou 26 segundos.

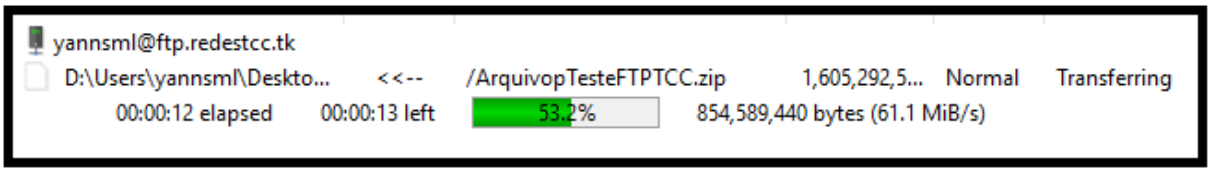
Figura 144 – Taxa de *Upload FileZilla WorkSpaces*



Fonte: alterado baseado em FileZilla (2021)

Ainda logado no *WorkSpaces* e FTP, foi solicitado o *download* do arquivo para a máquina do cliente *WorkSpaces*. Conforme ilustrado na Figura 145, a taxa de *download* teve em média 61.1 Mbps. Este *download* demorou 24 segundos.

Figura 145 – Taxa de *Download FileZilla WorkSpaces*



Fonte: alterado baseado em FileZilla (2021)

Com esses resultados, percebeu-se que houve um ganho de 46,4 Mbps em taxa de *upload* e 60,1 Mbps em taxa de *download* utilizando o aplicativo *WorkSpaces* ao invés do *OpenVPN*. Também houve de perda de 2300% do tempo em relação ao *upload* e 6316% em relação ao *download*.

Portanto, a utilização dos *WorkSpaces* proporciona desempenho, além de ser consideravelmente mais rápido em comparação com a utilização da VPN quando ocorre transferência de arquivos com o servidor FTP.

6 CONCLUSÃO

Este trabalho teve como objetivo responder a seguinte questão de pesquisa: Virtualização de servidores pode proporcionar usabilidade, confiabilidade e desempenho?

Portanto, usando a plataforma AWS, concluiu-se que é possível implementar a técnica de virtualização de servidores proporcionando usabilidade, confiabilidade e desempenho. Com esse trabalho, foi possível simular a implementação da técnica de virtualização de servidores na plataforma AWS.

Testando a segurança da rede percebeu-se que ela consegue lidar com várias solicitações ao mesmo tempo, sem perder a confiabilidade. Também foi observado que as instâncias criadas seguem as regras dessa rede. Essa plataforma também garante segurança extra para suas instâncias com as *Security Groups*.

Ao realizar testes para verificar o funcionamento dos componentes da rede observou-se que os servidores estão realizando suas ações conforme o especificado, garantindo a usabilidade da rede.

Testes foram realizados para verificar a velocidade das instâncias dos clientes, sendo observado que houve ganho de desempenho ao utilizar o serviço *WorkSpaces,* porém houve perda ao utilizar o serviço VPN.

Para continuidade deste trabalho sugere-se os seguintes trabalhos futuros:

- Implementar o servidor VPN utilizando outro serviço.
- Criar uma solução para que cada cliente tenha um usuário e que consiga autenticar em todos os servidores com esse usuário.
- Comparar a usabilidade, confiabilidade e desempenho da técnica de virtualização de servidores, com a técnica de implementação de servidores convencional.

7 REFERÊNCIAS

ALENCAR, Felipe. **Chrome: o navegador completo e gratuito do Google**. TechTudo. 2016. Disponível em: https://www.techtudo.com.br/tudo-sobre/google-chrome.html. Acesso em: 07 dez. 2020

Apache. **htpasswd - Manage user files for basic authentication**. Apache. 2020. Disponível em: https://httpd.apache.org/docs/2.4/programs/htpasswd.html. Acesso em: 07 dez. 2020.

AWS. **Amazon EC2: Capacidade de computação segura e redimensionável para oferecer suporte a praticamente qualquer carga de trabalho**. AWS. 2020. Disponível em: https://aws.amazon.com/pt/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc. Acesso em: 19 out. 2020.

AWS. **Amazon Relational Database Service (RDS).** AWS. 2021. Disponível em: https://aws.amazon.com/pt/rds/. Acesso em: 13 mar. 2021.

AWS. **Amazon Route 53: Uma forma confiável e econômica de encaminhar usuários finais para aplicativos de Internet.** AWS. 2020. Disponível em: https://aws.amazon.com/pt/route53/. Acesso em: 17 out. 2020.

AWS. **Amazon Virtual Private Cloud**. AWS. 2020. Disponível em: https://aws.amazon.com/pt/vpc/. Acesso em: 19 out. 2020.

AWS. **Amazon WorkSpaces: Acesse seu desktop de qualquer lugar, a qualquer instante, em qualquer dispositivo**. AWS. 2020. Disponível em: https://aws.amazon.com/pt/workspaces/?nc1=h_ls&workspaces-blogs.sort-by=item.additionalFields.createdDate&workspaces-blogs.sort-order=desc. Acesso em: 19 out. 2020.

AWS. **AWS Directory Service**. AWS. 2021. Disponível em: https://aws.amazon.com/pt/directoryservice/. Acesso em: 10 abr. 2021.

AWS. **Gateways da Internet.** AWS. 2020. Disponível em: https://docs.aws.amazon.com/pt_br/vpc/latest/userguide/VPC_Internet_Gateway.html. Acesso em: 22 out. 2020.

AWS. **Gerenciar contas de usuário na instância do Amazon Linux.** AWS. 2020. Disponível em: https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/managing-users.html. Acesso em: 30 out. 2020.

AWS. **Grupos de segurança para a VPC**. AWS. 2020. Disponível em: https://docs.aws.amazon.com/pt_br/vpc/latest/userguide/VPC_SecurityGroups.html. Acesso em: 21 out. 2020.

AWS. **Network ACLs**. AWS. 2020. Disponível em: https://docs.aws.amazon.com/pt_br/vpc/latest/userguide/vpc-network-acls.html. Acesso em: 23 out. 2020.

AWS. **O que é AWS ?.** AWS. 2020. Disponível em: https://aws.amazon.com/pt/what-is-aws/. Acesso em: 19 out. 2020.

AWS. **Perguntas frequentes sobre a Amazon VPC**. AWS. 2020. Disponível em: https://aws.amazon.com/pt/vpc/faqs/. Acesso em: 22 out. 2020.

AWS. **Tabelas de rotas**. AWS. 2020. Disponível em: https://docs.aws.amazon.com/pt_br/vpc/latest/userguide/VPC_Route_Tables.html. Acesso em: 22 out. 2020

AWS. **Trabalhar com a Amazon Virtual Private Cloud entre regiões**. AWS. 2020. Disponível em: https://docs.aws.amazon.com/pt_br/devicefarm/latest/developerguide/amazon-vpc-cross-region.html#device-farm-vpce-configuration-cross-region-step2. Acesso em: 23 out. 2020.

BRODBECK, Cassio. **Implantação de firewalls, confira as boas práticas**. Ostec. 2016. Disponível em: https://ostec.blog/seguranca-perimetro/implantacao-de-firewalls-boas-praticas#:~:text=A%20pol%C3%ADtica%20padr%C3%A3o%20nada%20mais,pode%20ser%20liberado%20ou%20bloqueado.. Acesso em: 23 out. 2020.

CLEMENTE, Matheus. **O que é host/provedor de hospedagem e quais são os melhores para o meu blog?.** RockContent. 2019. Disponível em: https://rockcontent.com/br/blog/host/. Acesso em: 22 out. 2020.

COSTA, Matheus Bigogno. **3 vantagens e desvantagens de usar o Firefox no PC.** Canaltech. 2020. Disponível em: https://canaltech.com.br/navegadores/firefox-vantagens-e-desvantagens/. Acesso em: 13 mar. 2021.

COSTA, Matheus Bigogno. **O que é Firewall.** iBest. 2020. Disponível em: https://canaltech.com.br/internet/o-que-e-firewall/. Acesso em: 29 set. 2020.

DIAS, Diego. **Video: Tabela de Roteamento**. Comutadores. 2018. Disponível em: https://www.comutadores.com.br/video-tabela-de-roteamento/#:~:text=A%20tabela%20de%20roteamento%20possui,OSPF%2C%20BGP%2Cetc).. Acesso em: 22 out. 2020.

DIAS, Luiz. **O que significa TI?.** Administradores.com. 2017. Disponível em: https://administradores.com.br/artigos/o-que-significa-ti. Acesso em: 25 set. 2020

FELIPE, Carlos. **O Que é PHP? Guia Básico de Programação PHP**. Hostinger. 2020. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-php-guia-basico. Acesso em: 13 mar. 2021.

FERNANDES, Rafael Silva; NUNO, Claudinei Di. **Virtualização de Servidores**. Revista Científica Multidisciplinar Núcleo do Conhecimento. v. 05, n .4,p. 34-44, abr. /2018. Disponível em: https://www.nucleodoconhecimento.com.br/tecnologia/virtualizacao-de-servidores?pdf=15002. Acesso em: 27 abr. 2021.

FERNANDES, Miriam. **Proxy: Tudo o que você precisa saber!.** STARTI. 2019. Disponível em: https://blog.starti.com.br/proxy/. Acesso em: 07 dez. 2020

FILIPE, Jeferson. **DMZ o que é e para que serve**. Falati. 2019. Disponível em: https://falati.com.br/dmz-o-que-e-e-para-que-serve/. Acesso em: 31 ago. 2020.

FOUROZAN, Behrouz A. **Comunicação de Dados e Redes de Computadores**. 3ª ed. Bookman. São Paulo. 2007. p. 631

GlobalFone. **O QUE É UM SOFTPHONE E QUAIS SÃO OS SEUS BENEFÍCIOS?.** GlobalFone. 2017. Disponível em: https://globalfone.com.br/blog/index.php/2017/10/23/softphone-e-seus-beneficios/. Acesso em: 21 out. 2020.

GHANNOUM, Rodrigo Gonçalves; RODRIGUES, Fábio Barbosa. **VIRTUALIZAÇÃO DE SERVIDORES: VANTAGENS E DESVANTAGENS**. REVISTA MIRANTE, Anápolis, v. 11,n. 6,p. 1-10, abr. /2018. Disponível em: https://www.revista.ueg.br/index.php/mirante/article/view/7612/5329. Acesso em: 27 abr. 2021.

GIL, Antônio Carlos. **Como Elaborar Projetos de Pesquisa**. 4. ed. São Paulo: ATLAS S.A., 2002.

GOMES, Vanessa. **Teste de desempenho de software.** TI Especialistas. 2015. Disponível em: https://www.tiespecialistas.com.br/teste-de-desempenho-de-software/#:~:text=Nos%20sistemas%20de%20computadores%20o,num%20determinado%20intervalo%20de%20tempo.. Acesso em: 1 out. 2020.

GUSTI, Yana. **TOP 20 ferramentas para teste de carga em 2018**. EasyQA. 2019. Disponível em: https://geteasyqa.com/pt/blog/best-tools-load-testing/. Acesso em: 02 mar. 2021.

HelpDigital. **O que é firewall?: Conceito, tipos e arquitetura**. HelpDigital. 2020. Disponível em: https://helpdigitalti.com.br/o-que-e-firewall-conceito-tipos-e-arquiteturas/. Acesso em: 21 out. 2020.

HUH, Jun‐Ho; SEO, Kyungryong. **Design and test bed experiments of server operation system using virtualization technology**. Human-centric Computing and Information Sciences. Busan, 2016. Disponível em: https://hcis-journal.springeropen.com/track/pdf/10.1186/s13673-016-0060-7. Acesso em: 31 ago. 2020.

ISKANDAR, Akbar; VIRMA, Elisabet; AHMAR, Ansari Saleh. **Implementing DMZ in Improving Network Security of Web Testing in STMIK AKBA**. Arxiv. Makassar, 2018. Disponível em: https://arxiv.org/ftp/arxiv/papers/1901/1901.04081.pdf. Acesso em: 31 ago. 2020.

JIANYUN, Chen; LiChunyan. **Research on meteorological information network security system based on VPN Technology .** MATEC Web of Conferences. Fuzhou, 2018. Disponível em: https://www.matec-conferences.org/articles/matecconf/pdf/2018/91/matecconf_eitce2018_01001.pdf. Acesso em: 29 set. 2020.

KRATZ, Edirlaine. **Porque utilizar um servidor de impressão na sua empresa?.** Helioprint. 2016. Disponível em: https://helioprint.com.br/blog/servidor-de-impressao/. Acesso em: 29 set. 2020.

LANDO, Andressa. **O que é ISO e por que certificar?.** Templum. 2019. Disponível em: https://certificacaoiso.com.br/o-que-e-iso-e-por-que-certificar/. Acesso em: 1 out. 2020.

LONGEN, Andrei. **Como Configurar o Cliente FileZilla**. Hostinger. 2020. Disponível em: https://www.hostinger.com.br/tutoriais/como-configurar-o-cliente-filezilla/#O-Que-e-FileZilla. Acesso em: 21 out. 2020.

LONGEN, Andrei**. O que é Apache? Uma Visão Aprofundada do Servidor Apache.** Hostinger. 2020. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-apache. Acesso em: 13 mar. 2021.

LONGEN, Andrei. **O Que É MySQL? Guia Para Iniciantes**. Hostinger. 2019. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-mysql. Acesso em: 13 mar. 2021.

LUCENA, Felipe. **Virtualização de servidores: o que é e como funciona.** Diferencial TI. 2016. Disponível em: https://blog.diferencialti.com.br/entenda-o-que-e-virtualizacao-de-servidores-e-como-funciona/. Acesso em: 13 set. 2020.

MACÊDO, Diego. **Introdução ao Wireshark: Detecção e captura de tráfego em redes.** Diego Macêdo. 2016. Disponível em: https://www.diegomacedo.com.br /introducao-ao-wireshark-deteccao-e-captura-de-trafego-em-redes/. Acesso em: 19 set. 2020.

MARQUES, Rafael. **O que é HTML? Entenda de forma descomplicada**. HomeHost. 2019. Disponível em: https://www.homehost.com.br/blog/tutoriais/o-que-e-html/. Acesso em: 13 mar. 2021

MASUDA, Hideo; SEGAWA, Shun; MORI, Masayuki. **Proposal and Prototype of DNS Server Firewall with Flexible Response Control Mechanism**. Atlantis press. Kyoto, 2019. Disponível em: https://www.atlantis-press.com/journals /ijndc/125925043/view. Acesso em: 31 ago. 2020.

MONTANARI, Luís. **Protocolo DHCP: O que é, como funciona e mais.** tiespecialistas. 2019. Disponível em: https://www.tiespecialistas.com.br/protocolo-dhcp-o-que-e-como-funciona-e-mais/. Acesso em: 31 ago. 2020.

MORAES, Daniel. **O que é URL e como ela é decisiva para o sucesso da sua estratégia digital**. Rockcontent. 2018. Disponível em: https://rockcontent.com/br/blog/url/. Acesso em: 31 ago. 2020.

MORELLATO, Fernando César. **VLANS: LANs Virtuais.** BLOG IPV7. 2018. Disponível em: https://www.blog.ipv7.com.br/tecnica/vlans-lans-virtuais/. Acesso em: 31 ago. 2020.

MUXFELDT, Pedro. **Local Área Network (LAN) - Rede local**. CCM. 2017. Disponível em: https://br.ccm.net/contents/255-local-area-network-lan-rede-local. Acesso em: 19 set. 2020

OGUNYEMI, Abiodun; JOHNSTON, Kevin. **Is Server Virtualization Implementation in Business and Public Organizations a Worthwhile Investment?.** World Scientific. Cidade do Cabo. Disponível em: https://www.worldscientific.com/ doi/pdf/10.1142/S0219622017500146. Acesso em: 19 set. 2020.

OpenVPN. **Primeiro login no acesso ao servidor Admin Web UI**. OpenVPN. 2020. Disponível em: https://openvpn.net/. Acesso em: 03 nov. 2020.

OpenVPN. **Uma VPN empresarial para acessar recursos de rede com segurança**. OpenVPN. 2020. Disponível em: https://openvpn.net/. Acesso em: 03 nov. 2020.

PEREIRA, Leonardo. **Qual a diferença entre internet e web?.** Olhar Digital. 2014. Disponível em: https://olhardigital.com.br/noticia/qual-a-diferenca-entre-internet-e-web/40770. Acesso em: 29 set. 2020.

PIMENTA, Rafael. **O que é Browser? Saiba mais sobre o assunto!.** Geek Blog. 2020. Disponível em: https://windowsteam.com.br/o-que-e-browser-saiba-mais-sobre-o-assunto/. Acesso em: 29 set. 2020.

PIMENTEL, Rafael. **O que é um sistema operacional?.** Geek Blog. 2020. Disponível em: https://windowsteam.com.br/o-que-e-um-sistema-operacional/. Acesso em: 18 out. 2020.

REIS, Fabio dos . **O que é Máscara de Sub-Rede**. Bóson treinamentos. 2017. Disponível em: http://www.bosontreinamentos.com.br/redes-computadores/o-que-e-mascara-de-sub-rede/. Acesso em: 17 out. 2020.

RUGGIERI, Ruggero. **Análise sobre a ISO 9126 : NBR 13596.** TI Especialistas. 2016. Disponível em: https://www.tiespecialistas.com.br/analise-sobre-iso-9126-nbr-13596/. Acesso em: 1 out. 2020.

RUWAIDA, Devi; KURNIA, Dian. **RANCANG BANGUN FILE TRANSFER PROTOCOL (FTP) DENGAN PENGAMANAN OPEN SSL PADA JARINGAN VPN MIKROTIK DI SMKS DWIWARNA.** E-Journal of Unimed. Medan, 2018. Disponível em: https://jurnal.unimed.ac.id/2012/index.php/cess/article/download/8267/7759. Acesso em: 29 set. 2020.

SALES, Vanessa de Alcântara. **Entenda o que são os Testes de Desempenho**. EasyQA. 2019. Disponível em: https://blog.cedrotech.com/entenda-o-que-sao-os-testes-de-desempenho/. Acesso em: 02 mar. 2021.

SALUTES, Bruno. **O que é IP**. Canaltech. 2019. Disponível em: https://canaltech.com.br/software/o-que-e-ip/. Acesso em: 31 ago. 2020.

SOUZA, Ivan de . **Banco de dados: saiba o que é, os tipos e a importância para o site da sua empresa**. RockContent. 2020. Disponível em: https://rockcontent.com/br/blog/banco-de-dados/. Acesso em: 13 mar. 2021.

SOUZA, Ivan de. **Entenda o que é HTTP e o quão importante esse protocolo é para o seu site.** Rockcontent. 2019. Disponível em: https://rockcontent. com/br/blog/http/. Acesso em: 31 ago. 2020.

SOUZA, Ivan de . **Saiba o que é ping e a relação dele com a sua velocidade de conexão**. RockContent. 2020. Disponível em: https://rockcontent.com/br/blog/ping/. Acesso em: 22 out. 2020.

Speedcheck. **Pacote**. Speedcheck. Disponível em: https://www.speedcheck.org/pt/wiki/pacote/. Acesso em: 17 out. 2020.

TAVARES, Lucas. **O que é um Servidor Web (Web Server).** Melhor hospedagem de sites. 2019. Disponível em: https://www.melhoreshospedagemdesites.com/ servidor-web/. Acesso em: 19 set. 2020.

TI-Forense. **MD5**. TI-Forense. 2018. Disponível em: https://www.tiforense.com.br/md5/. Acesso em: 07 dez. 2020.

TRIPATHI, Nikhil; HUBBALLI, Neminath. **Detecting Stealth DHCP Starvation Attack using Machine Learning Approach.** researchgate. Madhya Pradesh, 2017. Disponível em: https://www.researchgate.net/publication/320558117_Detecting _Stealth_DHCP_Starvation_Attack_using_Machine_Learning_Approach. Acesso em: 31 ago. 2020.

VARELLA, Claudia. **CPF: O que é, para que serve e como tirar**. UOL. 2019. Disponível em: https://economia.uol.com.br/noticias/redacao/2019/12/11/cpf-o-que-e-para-que-serve-como-tirar.htm. Acesso em: 19 set. 2020.

VOLPATO, Elisa . **O que é usabilidade?.** Medium. 2016. Disponível em: https://medium.com/testr/o-que-%C3%A9-usabilidade-579f9b285d8e. Acesso em: 1 out. 2020

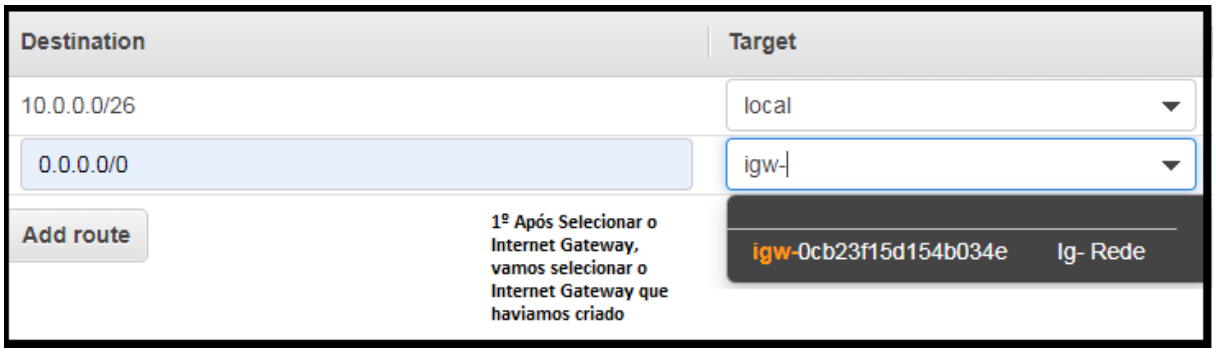
WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação**. 2. ed. Rio de Janeiro: Elsevier Editora Ltda., 2014.

ANEXO A – ACESSO A INTERNET DE MÁQUINAS PRIVADAS

As instâncias que estão localizadas na sub-rede privada necessitam de acesso à *Internet* periodicamente para baixar os recursos do servidor ou atualizações. Porém, para isso, é necessário criar temporariamente rotas da sub-rede privada para a *Internet* e flexibilizar temporariamente as regras de entrada e saída do *firewall* e da *Security Group*.

Para obter esse acesso, foi alterada a tabela de roteamento privado, conforme ilustrado na Figura 146, criando uma rota da sub-rede privada para a *Internet,* utilizando o *Internet Gateway* que a sub-rede pública usa para acesso à *Internet*.

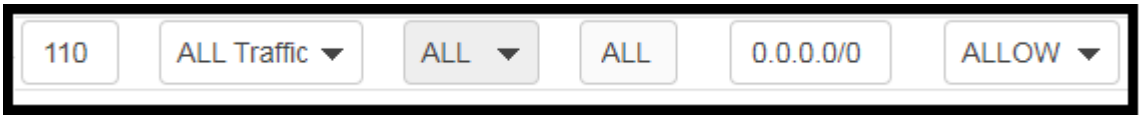
Figura 146 – Criar rota de *Internet* sub-rede privada



Fonte: alterado baseado em AWS (2020)

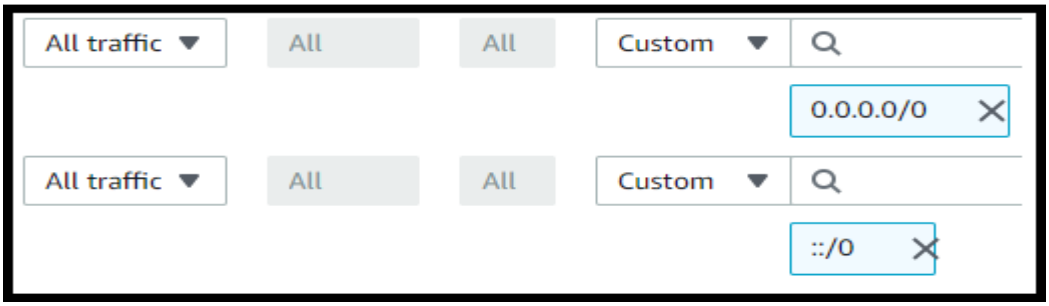
É adicionada a regra de permissão de entrada e saída de qualquer porta e de qualquer IP na ACL Privada, conforme mostrado na Figura 147. Lembrando que é necessário adicionar essa regra nas regras de entrada e depois nas regras de saída da ACL privada. Na *Security Group* do servidor que deseja ter acesso à *Internet* é adicionada a regra de entrada e saída de todo tráfego, vinda qualquer porta e de qualquer IP, conforme mostrado na Figura 148. Semelhante as regras da ACL, deve-se incluir as regras de entrada e saída separadamente.

Figura 147 – Regras de entrada e saída ACL Privada *Internet*



Fonte: alterado baseado em AWS (2020)

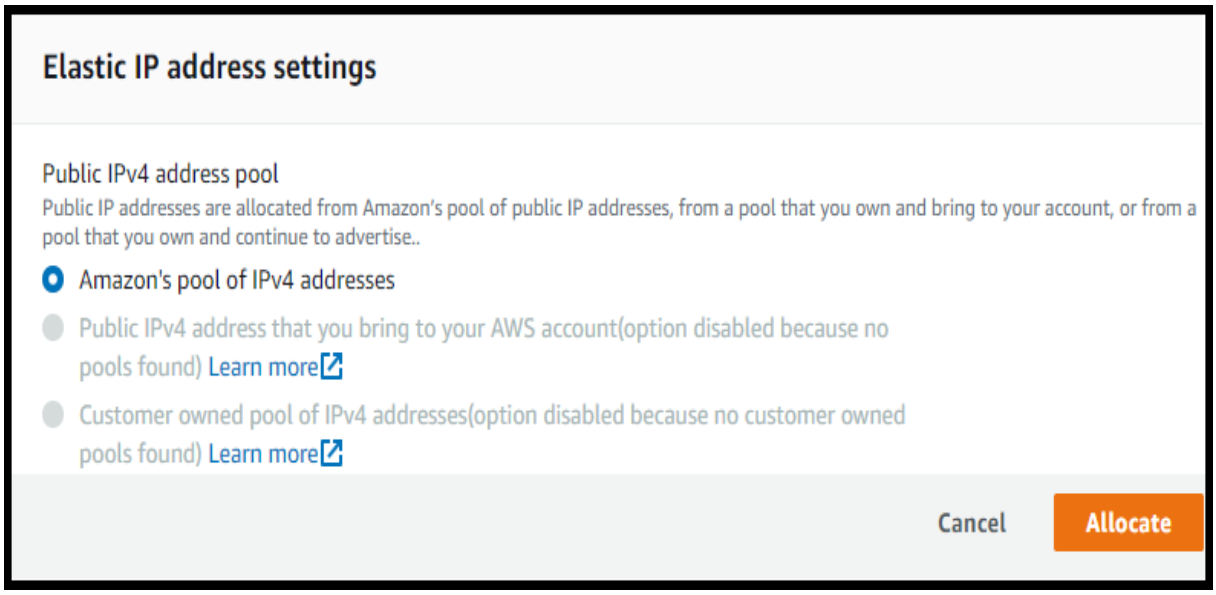
Figura 148 – Regras de entrada e saída *Security Group Internet*



Fonte: alterado baseado em AWS (2020)

Agora é necessário associar um IP público ao servidor. Para isso foi acessada a aba "*Elastic IPs"* e foi alocado um endereço de IP público conforme Figura 149.

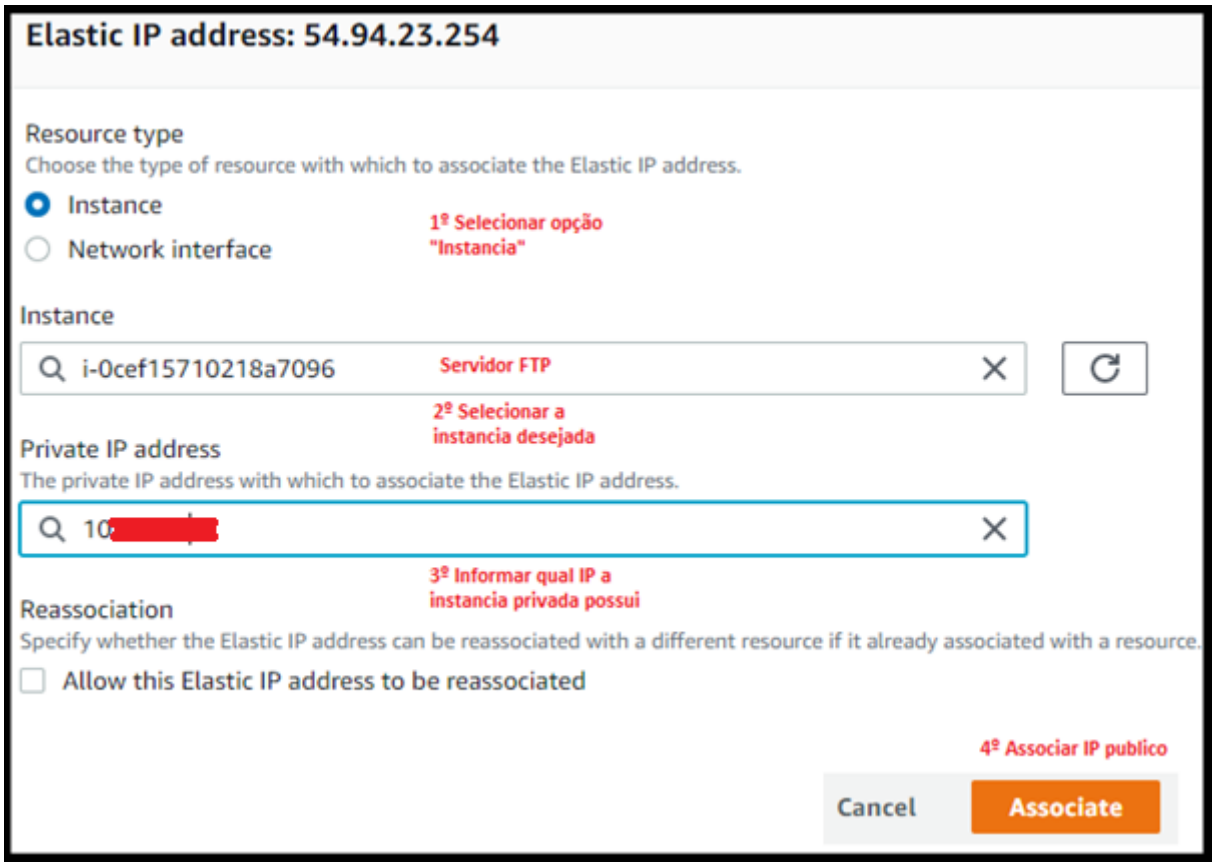
Figura 149 – Alocar IP público



Fonte: alterado baseado em AWS (2020)

Esse IP público deve ser associado ao servidor que deseja ter acesso temporário a *Internet* como, por exemplo, o servidor FTP conforme ilustrado na Figura 150. É informado que este IP deve ser associado a uma instância, é informado qual instância e qual o IP privado dessa instância. Após essa configuração a instância tem acesso à *Internet*.

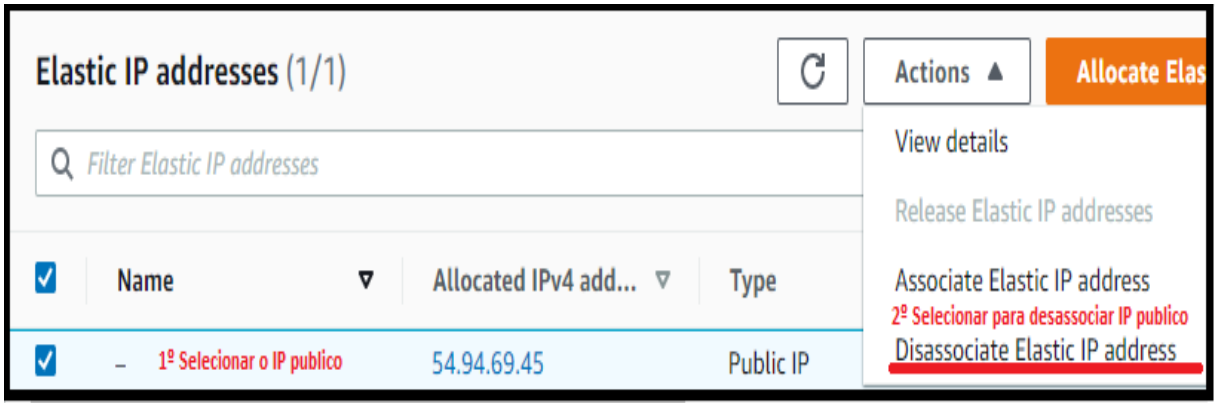
Figura 150 – Associar IP público instância privada



Fonte: alterado baseado em AWS (2020)

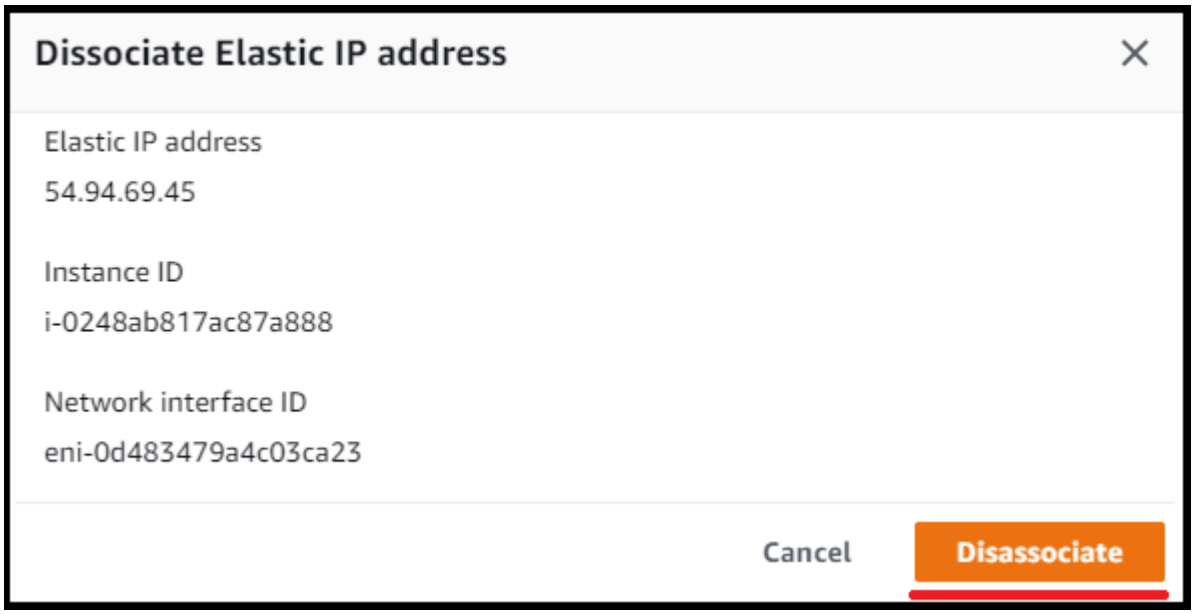
Após finalizar as solicitações que necessitam da *Internet*, é necessário que o servidor volte a não possuir acesso à *Internet*. Então, foi desassociado o IP público da instância conforme ilustrado na Figura 151 e na Figura 152. A seguir, conforme ilustrado na Figura 153, é liberado o IP público.

Figura 151 – Opção desassociar IP público



Fonte: alterado baseado em AWS (2020)

Figura 152 – Desassociar IP público



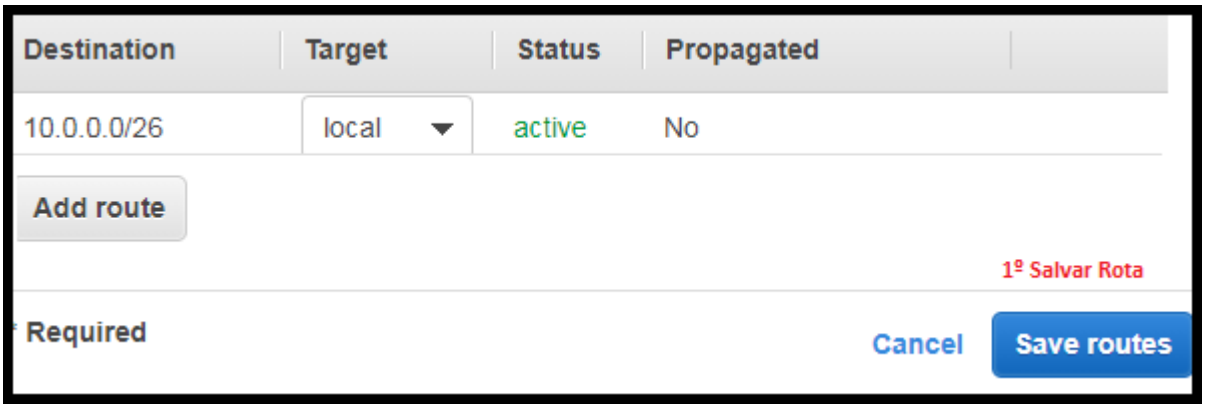
Fonte: alterado baseado em AWS (2020)

Figura 153 – Liberar IP público

Fonte: alterado baseado em AWS (2020)

Foi acessada a tabela de roteamento privado e excluído a rota que utilizava o *Internet Gateway* para acesso à *Internet*. Essa tabela ficou da forma mostrada na Figura 154.

Figura 154 – Retornar padrão rotas privadas



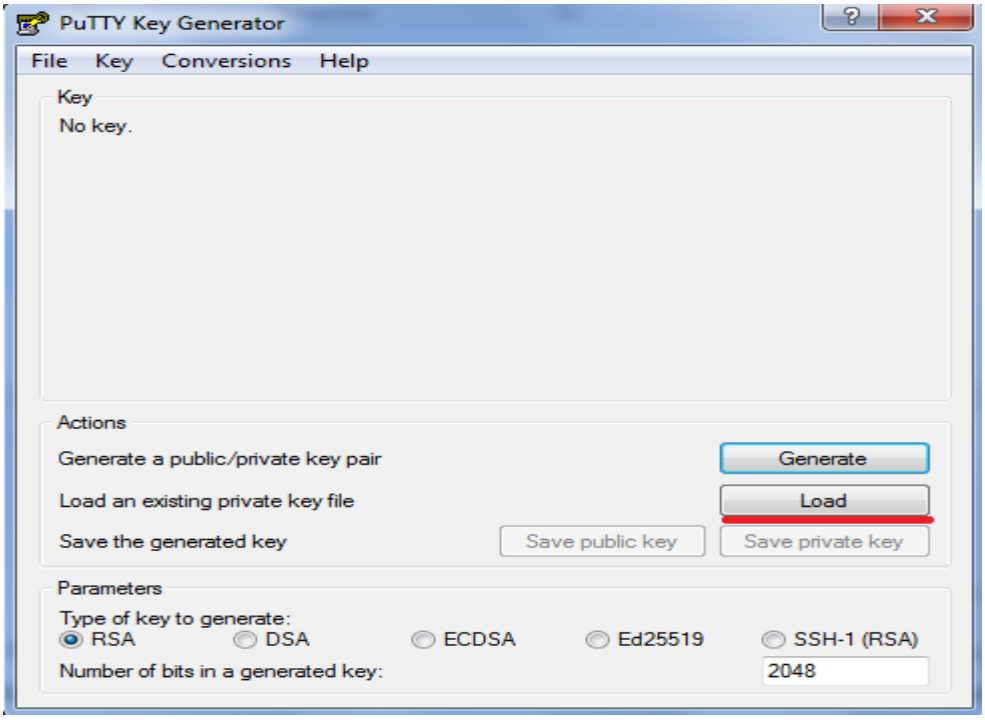
Fonte: alterado baseado em AWS (2020)

A seguir, foram excluídas as regras de entrada que liberavam tráfego de qualquer porta e de qualquer IP, e as regras de saída que liberavam tráfego de qualquer porta e de qualquer IP da ACL privada. Por fim, foram excluídas as regras de entrada que liberavam tráfego de qualquer porta e de qualquer IP, e as regras de saída que liberavam tráfego de qualquer porta e de qualquer IP da *Security Group* do servidor que possuía *Internet.* Após essas configurações, o servidor deixou de ter acesso à *Internet.*

ANEXO B – ACESSO SSH A INSTÂNCIA *OPENVPN* E UBUNTU

Para acesso a instâncias Ubuntu e *OpenVPN* foi utilizado o *software* PuTTY. Foi utilizada a versão 0.74 de 64 *bits* do *Windows*. Como mostrado na Figura 155, foi acessado o aplicativo PuTTYgen para geração de chave privada em formato que o PuTTY conhece. Com isso, foi acessada a opção "*Load*" para carregar a chave privada que foi gerada ao criar instância.

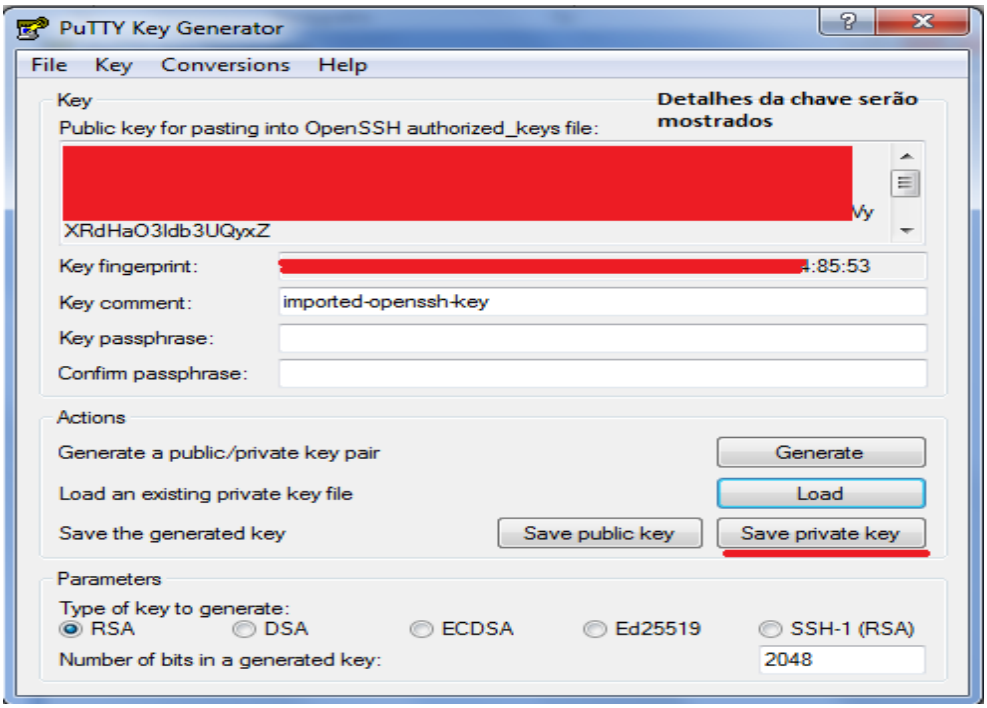
Figura 155 – Carregar chave privada



Fonte: alterado baseado em PuTTY (2020)

Conforme mostrado na Figura 156, após carregar a chave privada da instância o aplicativo PuTTYgen apresenta informações sobre a chave privada. Então, selecionada a opção "*Save private key*", foi escolhido local para salvar essa nova chave que o PuTTY consegue decifrar.

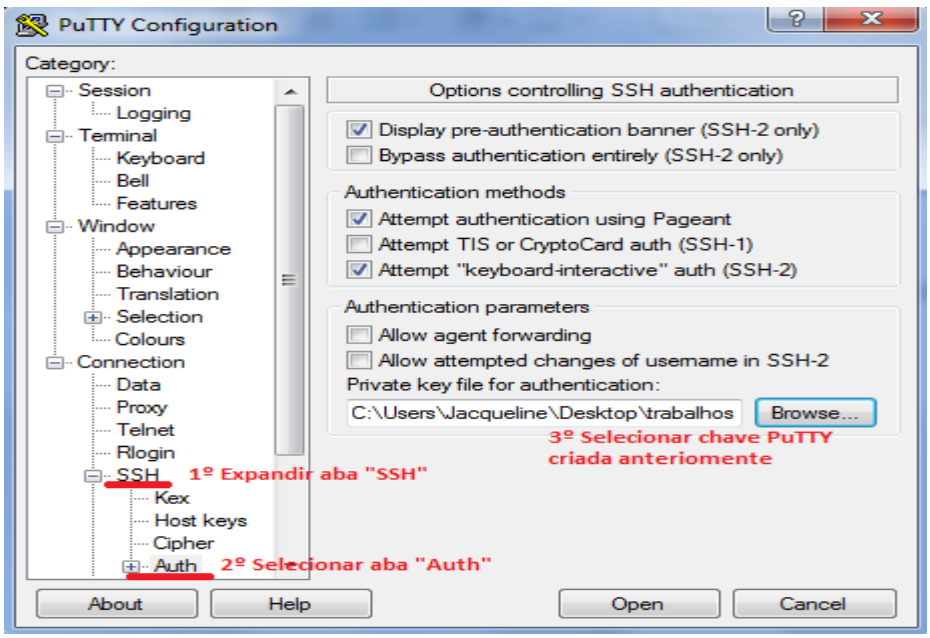
Figura 156 – Criação chave PuTTY



Fonte: alterado baseado em PuTTY (2020)

Foi aberto aplicativo *PuTTY* e conforme mostrado na Figura 157, foi expandida a aba “SSH”, selecionada a aba “Auth” e importada a chave PuTTY que foi criada anteriormente.

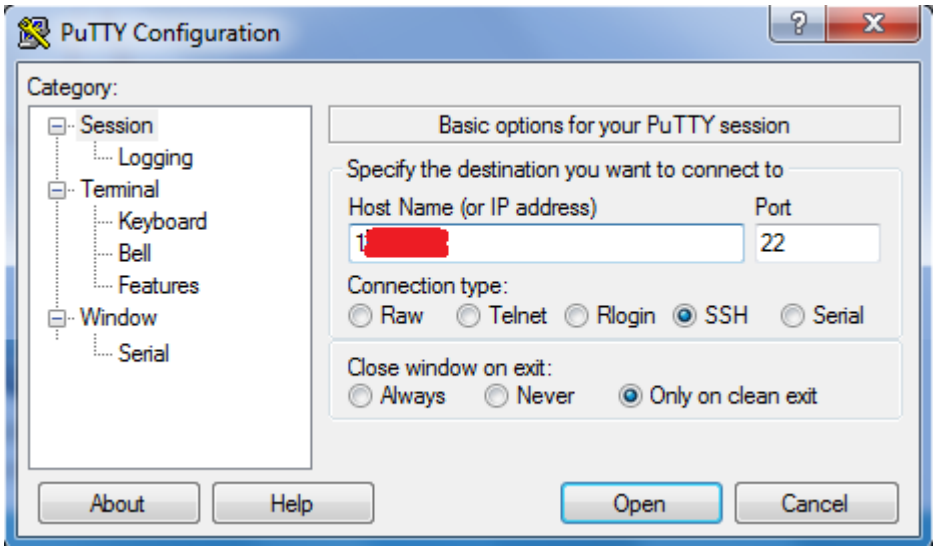
Figura 157 – Seleção chave PuTTY



Fonte: alterado baseado em PuTTY (2020)

Na instância *OpenVPN* não é necessário informar o nome do usuário. Portanto, é necessário informar somente o IP da instância conforme mostrado na Figura 158. Após essa ação possuímos acesso a instância.

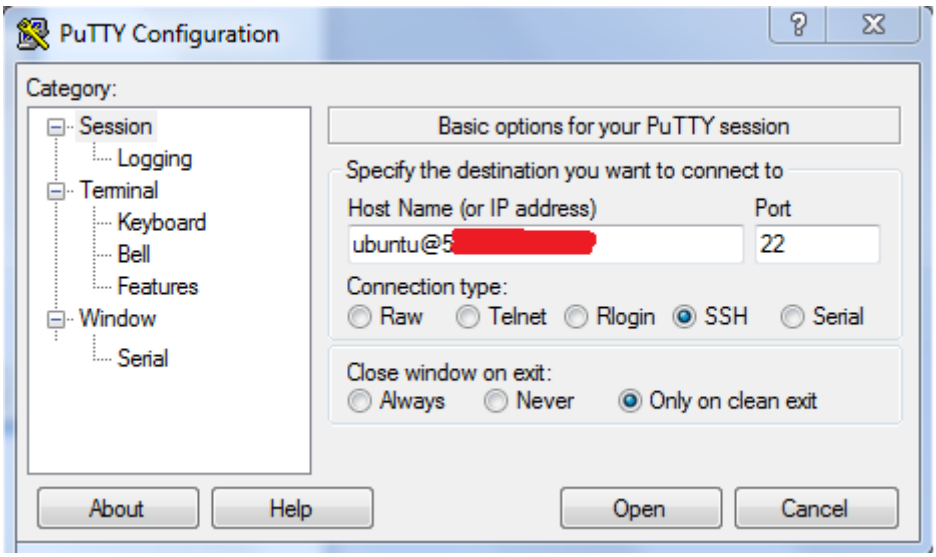
Figura 158 – Acesso a instância *OpenVPN*



Fonte: alterado baseado em PuTTY (2020)

Para acesso a instâncias Ubuntu na plataforma AWS é necessário *logar* com o usuário “ubuntu”. A instância Ubuntu é acessada pelo PuTTY, informando o nome do usuário, @ e IP da instância conforme mostrado na Figura 159. Após essa etapa é possível ter acesso a instância (AWS,2020).

Figura 159 – Acesso a instância Ubuntu

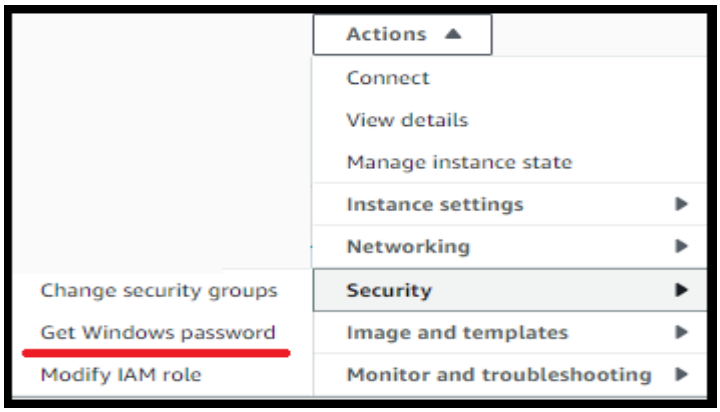


Fonte: alterado baseado em PuTTY (2020)

ANEXO C – ACESSO RDP A INSTÂNCIA WINDOWS

Para acesso a instâncias *Windows* foi utilizado o *software* nativo do *Windows* “Conexão de Área de Trabalho Remota”. Conforme mostrado na Figura 95, na página EC2, foi selecionada a opção “*Actions*”, selecionada a aba “*Security*” e escolhida a opção “*Get Windows password*”.

Figura 160 – Acesso para geração senha RDP



Fonte: alterado baseado em AWS (2020)

Conforme mostrado na Figura 161, é importada a chave privada baixada ao criar instância e após isso é apresentado informações sobre essa chave. Então, foi selecionada a opção “*Decrypt Password*” para decriptar essa chave.

Figura 161 – Importação chave privada RDP

Fonte: alterado baseado em AWS (2020)

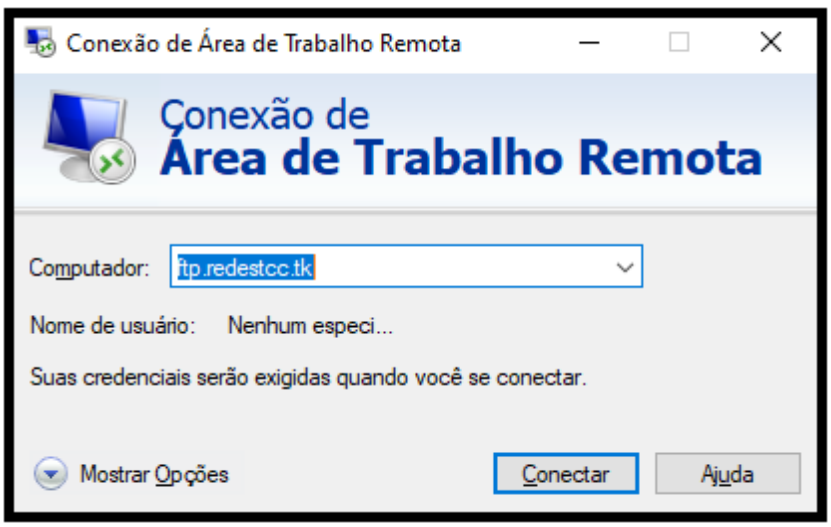
Após finalizar a etapa anterior, conforme mostrado na Figura 162, é apresentado o IP da instância, o nome de usuário e a senha. Foi copiado usuário e senha para etapa posterior.

Figura 162 – Geração senha decriptografada

Fonte: alterado baseado em AWS (2020)

Como mostrado na Figura 163, é aberto o aplicativo “Conexão de Área de Trabalho Remota” e informado o nome que roteia o IP da instância que deseja acessar. Nesse exemplo, foi utilizado o nome do servidor FTP.

Figura 163 – Iniciar conexão remota



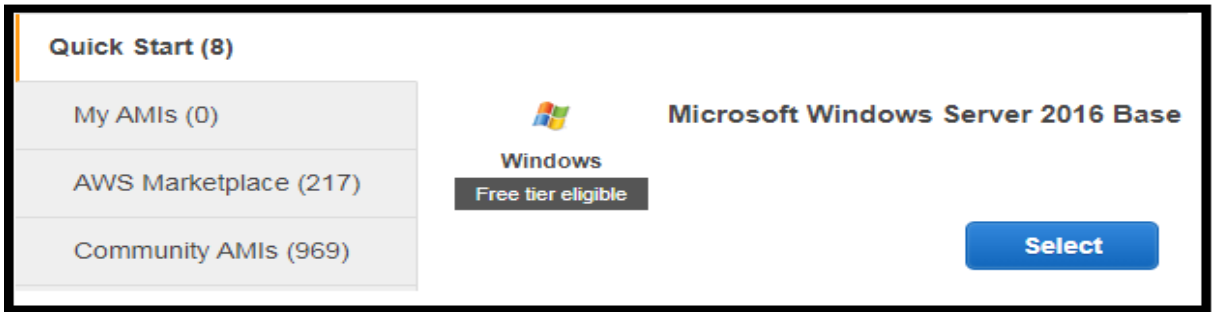
Fonte: alterado baseado em Windows 7(2020)

Foram informados o usuário e a senha, copiados anteriormente, e então, é acessada a instância.

ANEXO D – CRIAÇÃO INSTÂNCIA WINDOWS E UBUNTU

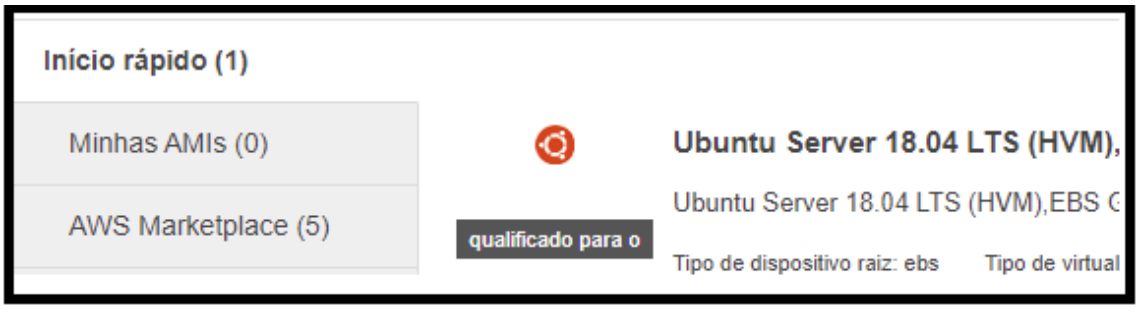
Para a criação de uma instância *Windows*, foi escolhida uma AMI "*Microsoft Windows Server* 2016 *Base*" que possui o SO *Windows Server* 2016, conforme ilustrado na Figura 164. Ou pode ser escolhida a AMI "Ubuntu Server 18.04 LTS (HVM)", que possui o SO Ubuntu Server 18.04, conforme apresentado na Figura 165.

Figura 164 – Escolha AMI *Windows*



Fonte: alterado baseado em AWS (2020)

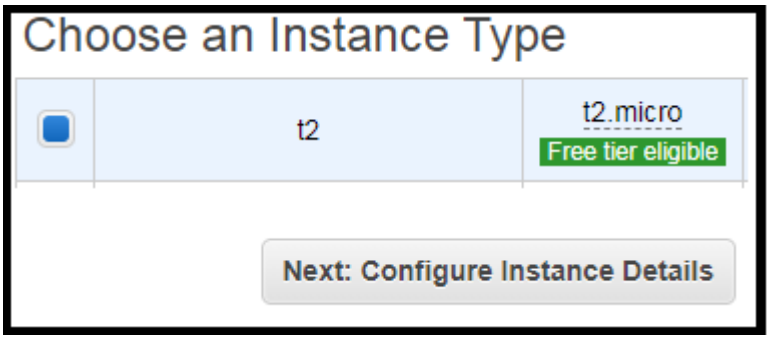
Figura 165 – Escolha AMI Ubuntu



Fonte: alterado baseado em AWS (2020)

Foi escolhida a instância do tipo "t2.*micro*" devido a sua gratuidade, conforme mostrado na Figura 166.

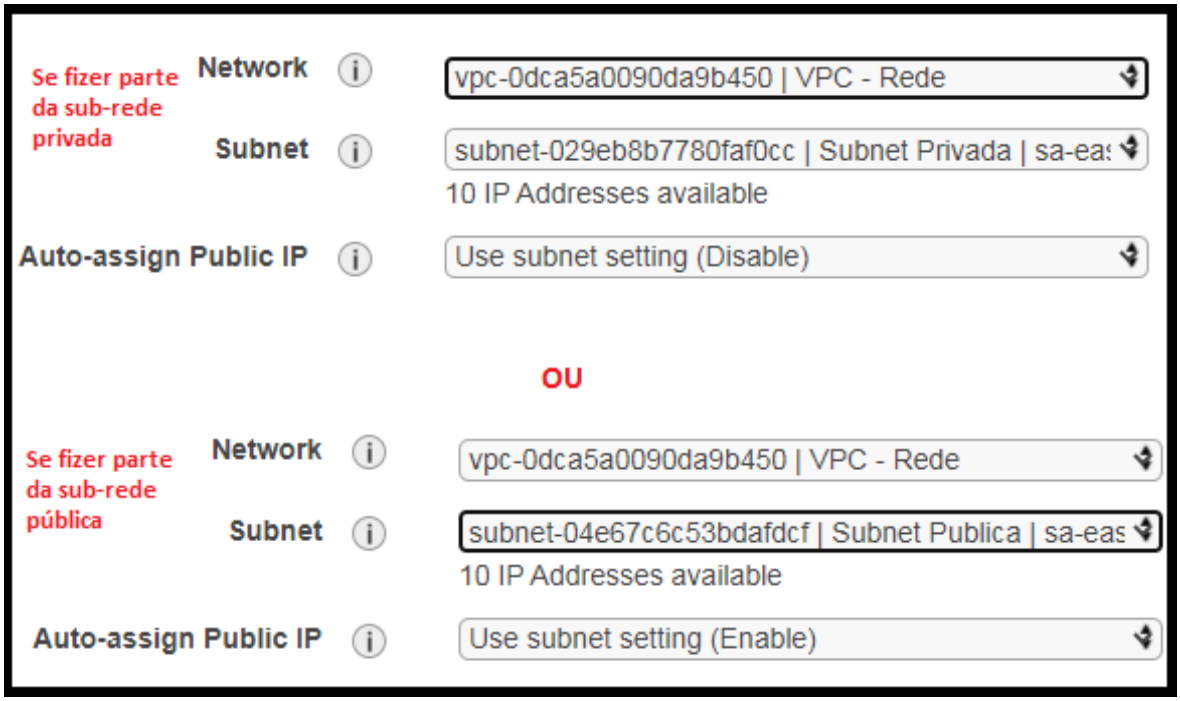
Figura 166 – Escolha tipo de instância



Fonte: alterado baseado em AWS (2020)

É escolhida a rede “VPC-Rede” e é escolhida a sub-rede, se ela faz parte da sub-rede pública ou sub-rede privada, conforme mostrado na Figura 167.

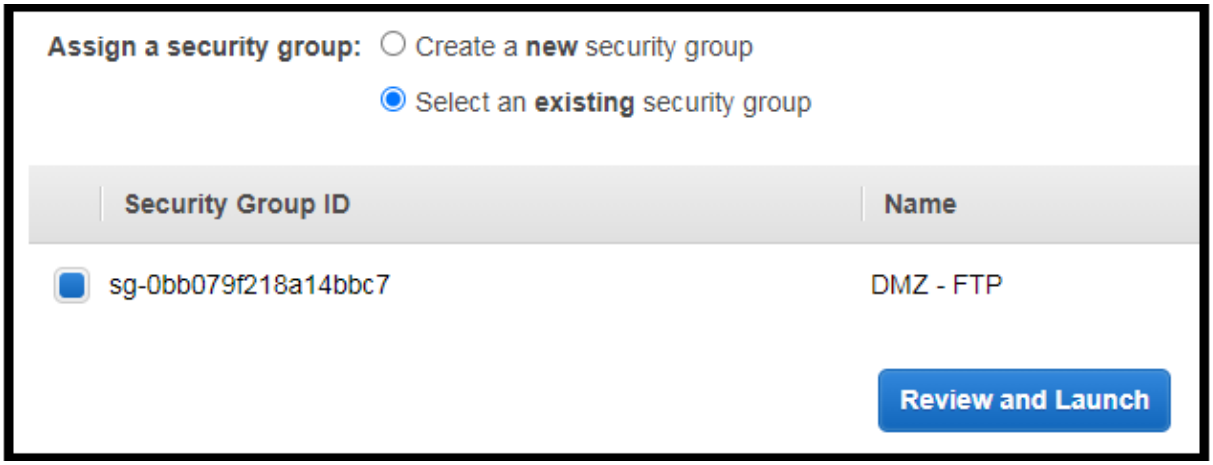
Figura 167 – Escolha rede e sub-rede



Fonte: alterado baseado em AWS (2020)

É selecionada a *security group* referente ao servidor que está sendo criado, como por exemplo, o servidor FTP que está ilustrado na Figura 168.

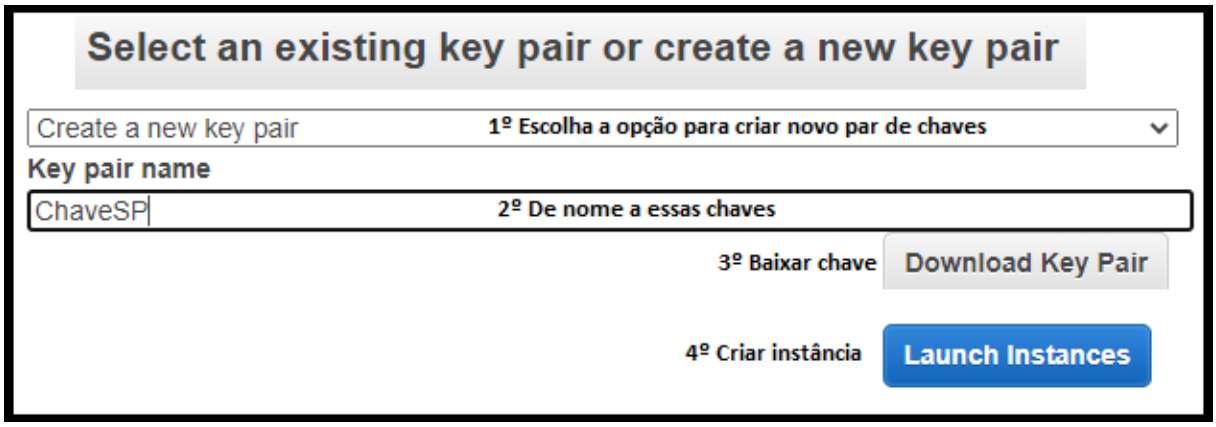
Figura 168 – Seleção *Security Group*



Fonte: alterado baseado em AWS (2020)

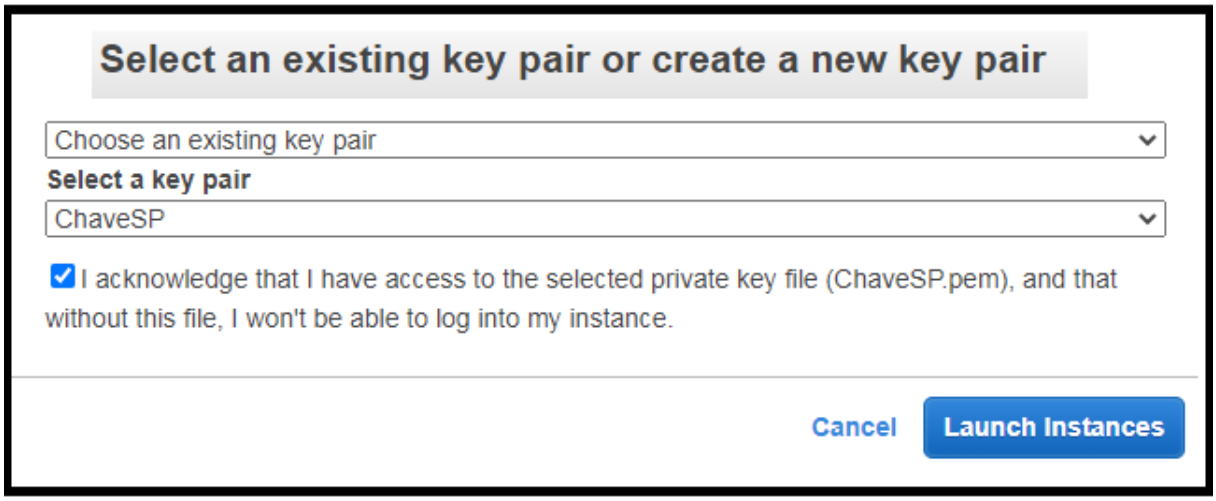
Se for necessário, criar par de chaves conforme ilustrado na Figura 169, senão informar um par de chave já criado, conforme ilustrado na Figura 170 e, então, finalizar a criação da instância.

Figura 169 – Criação de par de chaves



Fonte: alterado baseado em AWS (2020)

Figura 170 – Escolha de par de chaves



Fonte: alterado baseado em AWS (2020)