

INSTITUTO INFNET
ESCOLA SUPERIOR DE TECNOLOGIA
RDC – GRADUAÇÃO EM REDES DE COMPUTADORES



**Projeto de Bloco: Arquitetura de Infraestrutura de
Aplicações – 21GRLRDC01BAA501 – 21E3_5**

Aluno: Marcinei Zimmermann
E-mail: marcinei.zimmermann@al.infnet.edu.br
Github: https://github.com/marcineizimmermann/PB_Arquitetura
Professor: Fabiano Alves Gisbert
Data de Entrega: 07/12/2021

Sumário

1 Introdução.....	3
1.1 Objetivos.....	3
2 A Empresa.....	4
2.1 Requisitos.....	5
3 Detalhamento do Projeto.....	6
3.1 Aplicações.....	7
3.2 Arquitetura de Software.....	10
3.3 Recursos.....	11
3.4 Virtualização.....	14
3.5 Plano de Implementação.....	15
4 Metodologia.....	15
4.1 Recursos.....	16
4.2 Considerações.....	18
5 Arquitetura.....	18
6 Requisitos Técnicos.....	19
7 Implementação.....	24
7.1 Modelo de Simulação.....	24
7.2 Arquitetura.....	25
7.3 Metodologia.....	26
7.4 Hardware.....	26
7.5 Topologia.....	27
7.6 Virtualizador.....	28
7.7 Plano de Implantação.....	30
7.7.1 Virtualizador.....	32
7.7.2 Rede.....	42

7.7.3 Armazenamento.....	48
7.7.4 Sistemas Operacionais.....	61
7.7.5 Ansible.....	71
7.7.6 Docker.....	77
7.7.7 Código Fonte.....	78
7.7.8 Pós configuração.....	86
7.8 Testes.....	88
8 Análise da Estrutura.....	94
9 Conclusão.....	95
Link Apresentação.....	97
Link Github.....	97
Bibliografia.....	98

1 Introdução

O presente documento visa apresentar o desenvolvimento de uma aplicação onde a sua infraestrutura está sendo utilizado em estrutura simplificada e com poucos recursos técnicos. Será utilizado esta demanda como objeto do estudo e implementação de solução em virtualização em servidor da própria empresa.

O virtualizador selecionado é o Vmware vSphere ESXi 7, sendo implantado diretamente no hardware de equipamento compatível com esta solução. A decisão foi tomada considerando que já é utilizado este produto, sendo necessário portanto, a implementação de novos conceitos e ferramentas a fim de se obter resultados positivos.

Além de buscar desempenho, também será almejado recursos de redundância e recuperação, desta forma oferecendo confiabilidade e compromisso com o funcionamento da aplicação.

A aplicação em questão, será o fornecimento de um portal na internet com informações e recursos para interação com os clientes da empresa, onde para isto será utilizado o Wordpress.

Para a implementação de toda a estrutura será utilizado o Ansible como automatizador das funções e rotinas de configuração, criação e ação dos objetos necessários. Lembrando que fazem parte deste escopo de execução, orquestrações do virtualizador, sistema operacional, aplicações e até mesmo configurações específicas.

1.1 Objetivo

A realização de maneira coordenada e seguindo padrões de segurança e estabilidade, serão as metas para o final do presente projeto. Por possuir inúmeros aspectos, ferramentas e processos, a complexidade da implantação e criação requer grande aplicação de estudos e busca por novos conhecimentos. Em relação ao resultado final, o site deve atender as necessidades de atendimento, buscando oferecer aos seus clientes novas formas de comunicação e interação.

Nas questões técnicas, a principal meta é poder oferecer instrumental capaz de automatizar a implantação da estrutura proposta. Como a gama de elementos envolvidas vai do virtualizador, passa pelo sistema operacional, aplicações e inclusive configurações específicas, cabe extremo comprometimento para interpretar e conhecer o uso do Ansible.

A função de gerenciar e implementar este processo é característica do profissional que atua com Devops. Este termo tem como significado descrever conjunto de práticas para integrar e coordenar o desenvolvimento da aplicação, implantação e suporte.

Pela visão administrativa é necessário se ater a aspectos financeiros e práticos, onde o resultado não deve exceder a realidade da empresa. Mas em contrapartida o funcionamento do novo portal trará visibilidade comercial, por consequência resultados financeiros e novos recursos operacionais ao atendimento e a operação.

2 A Empresa

A Digicomp atua na área de soluções de tecnologia da informação desde 1999, oferecendo serviços de estrutura de informática para empresas de diversos ramos de atividade. O objetivo da empresa é prestar serviços e entregar equipamentos que sejam ideais para o funcionamento do parque tecnológico, buscando dar suporte com ética e responsabilidade.



Figura 1 – Digicomp Informática

Está integrado na cultura da empresa a necessidade de busca por tecnologias inovadoras e contar com fornecedores e produtos compatíveis com a necessidade dos seus clientes. As soluções de redes de computadores é o braço de serviços que conta com a maior atenção por parte da equipe técnica, oferecendo o compromisso com o funcionamento ideal e constante.

2.1 Requisitos

O foco da aplicação é fornecer portal website que seja eficiente visualmente e operacional. Para isto será utilizado o Wordpress como gerenciador de conteúdo para a internet, baseado em PHP e com o banco de dados MySQL.

O site deve fornecer acesso as informações comerciais da empresa, recursos de suporte e links para ferramentas utilizadas entre a Digicomp e os seus clientes. Poder utilizar a estrutura é altamente indicado, pois irá permitir que os investimentos neste portal sejam os menores possíveis.

O objetivo do projeto é ser o primeiro passo para o crescimento progressivo e organizado do portal, sempre levando em consideração a demanda dos clientes. Esta crescente de demandas em acesso e recursos é decorrente da expansão e da credibilidade que a empresa tem adquirido entre os seus clientes e as pessoas responsáveis por confiar e indicar estes serviços.

No mesmo foco de evitar investimentos iniciais, também é priorizado a utilização de softwares e aplicações que são desenvolvidos por comunidades de software livre. Também será considerado a confiabilidade destas soluções, haja visto que é necessário respeitar o uso fundamental na estrutura da empresa.

Outro recurso importante é a criação de ferramentas de publicação de informações úteis aos clientes, orientações técnicas, apresentações dos produtos e as novidades tecnológicas que surgem ou que são comercializadas.

3 Detalhamento do Projeto

Nas etapas seguintes será apresentado a estrutura ideal para o funcionamento da aplicação em ambiente produtivo e com recursos de redundância e eficiência, haja visto a utilização por parte dos clientes da empresa como uma ferramenta rotineira de busca por informações e recursos.

Ao final deste desenvolvimento será possível verificar a intenção de que a solução esteja sendo oferecida de maneira segura, que a empresa possa realmente conta com os recursos alcançado e que os clientes utilizem como o meio fundamental de acesso a Digicomp.

Para o desenvolvimento do projeto é preciso definir as soluções que irão fazer parte da estrutura, desde as aplicações até a metodologia a ser utilizada para realização da implantação. Os aspectos físicos também serão demonstrados, haja visto que as suas características também irão guiar as escolhas das soluções.

A escolha feita pela estrutura de virtualização no próprio ambiente da empresa foi orientada pela presença desta estrutura já montada e operacional. Foi considerado que o investimento inicial para implantação é menor do que o valor final de um contrato de utilização de servidores em nuvem.

O planejamento é de que a empresa num período de 01 (um) ano terá crescimento suficiente para que a implantação da estrutura ideal seja realizada com recursos próprios ou financiamento bancário da diferença do valor necessário. Neste caso o financiamento, caso haja necessidade, e o valor utilizado pelo próprio caixa da empresa, serão considerados desde já para diluir o impacto destes sobre o fluxo de caixa.

Como mencionado anteriormente, o servidor atual da empresa comporta a utilização inicial da aplicação para a quantidade atual de clientes, acrescido de um crescimento médio característico de períodos anteriores a este projeto. Segue resumo da estrutura atual da empresa:

Equipamento	Descrição
Servidor	<ul style="list-style-type: none"> • AMD Ryzen 5 2600X 12 vCpu's 3.6GHz • Memória 32GB • Armazenamento vSphere: SSD 120GB • Datastore Primário: 2 x 2TB (raid 1) • Array Interno: 3 x 2TB (raid 5) • Placa de rede Intel Quad Port Gigabit
Storage	<ul style="list-style-type: none"> • Intel I3-3220 4 vCpu's 3.3GHz • Memória 6GB • Armazenamento vSphere: SSD 120GB • Array Interno: 3 x 4TB (raid 5) • Placa de rede Intel Dual Port Gigabit
Switch	<ul style="list-style-type: none"> • Intelbras SG5204MR 48 portas Gigabit L2
Nobreak	<ul style="list-style-type: none"> • NHS 2500VA Senoidal

Tabela 1 – Equipamentos estrutura atual

Definido a estrutura e a forma de viabilizar a implantação e crescimento, é possível especificar e definir as questões de software e recursos a serem utilizados.

3.1 Aplicações

Para a implantação da estrutura será utilizado a ferramenta de orquestração Ansible, que consiste num projeto de gerenciamento, automatização, configuração e implantação de equipamentos, sistemas operacionais, aplicações e até ações na estrutura.

O Ansible é executado de um servidor que poderá implementar em cada servidor da estrutura as seguintes funcionalidades e recursos:

- Virtualizador do tipo 1 (hypervisor);
- Máquinas virtuais;
- Configurações e aplicações específicas;
- Criação e configuração de containers;
- Configurações pós instalação;

Também é possível a configuração de equipamentos de rede, como switchs, roteadores, Access Points, entre outros.

Na imagem abaixo está representada a topologia básica da Aplicação, apresentando os equipamentos e suas respectivas funções, bem como suas ligações. Segue:

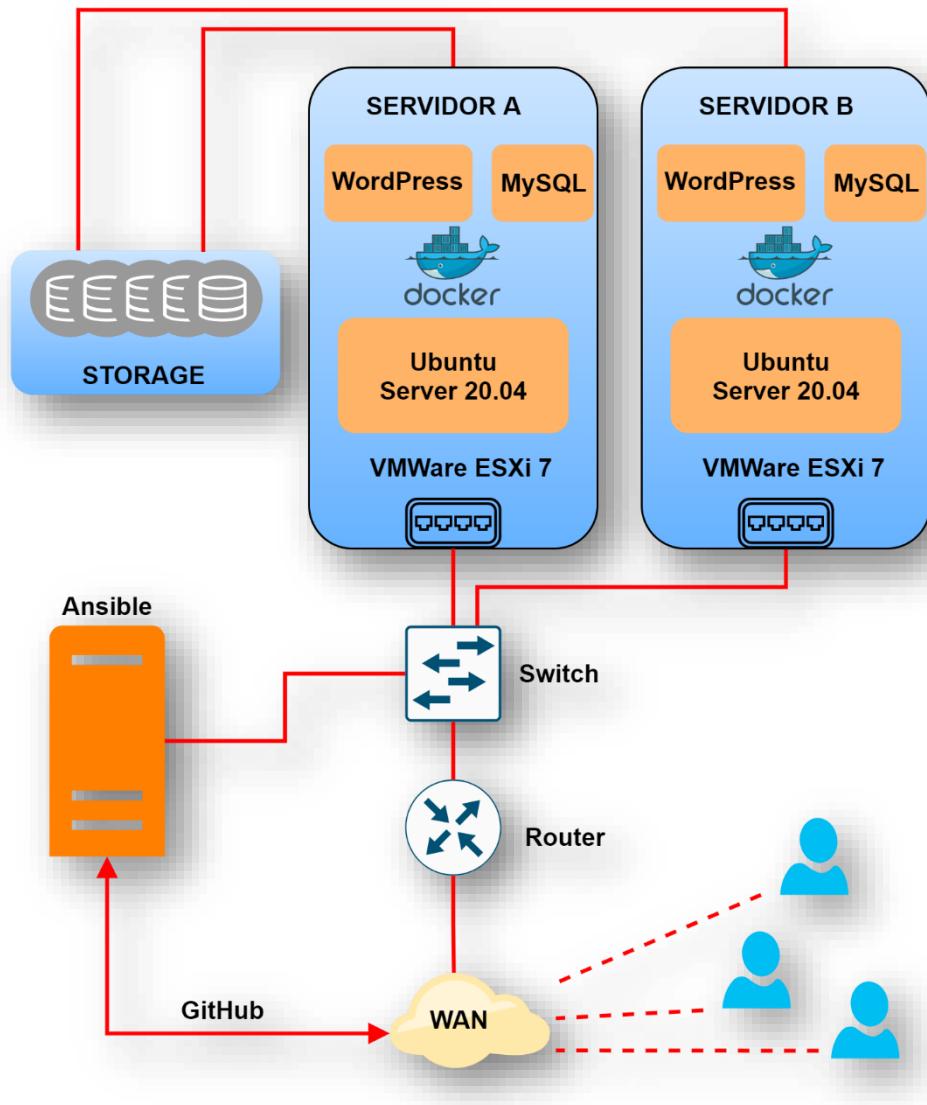


Figura 2 – Topologia Básica Aplicação

Como virtualizador do tipo 1 (bare metal), foi selecionado o Vmware, haja visto que atualmente já é utilizado esta solução na empresa, mas na sua versão gratuita. Para a nova estrutura será necessário realizar a troca por versão com recursos corporativos. As máquinas virtuais que irão receber as funções da aplicação terão implantados o sistema operacional Linux Ubuntu Server em sua versão 20.04, onde estarão os containers da aplicação e do banco de dados.

Para o controle dos códigos fonte será utilizado o GitHub, que consiste numa plataforma web onde códigos fontes e projetos ficam disponíveis abertamente ou de maneira controlada

para o compartilhamento de informações. Outra aplicação muito utilizada é o controle de implementações, correções e o histórico em si do projeto.

No projeto esta ferramenta será fundamental para o desenvolvimento da aplicação, possibilitando o acompanhamento das etapas e nos casos necessários as devidas correções.

3.2 Arquitetura do Software

O desenvolvimento da aplicação se dará no WordPress, que consiste num Sistema de Gerenciamento de Conteúdo – CMS (Content Management System), ou melhor, um gerenciador de sites de código aberto contando com uma comunidade ativa e disposta a manter as inovações constantes desta ferramenta.



Figura 3 – Componentes do WordPress

O Sistema WordPress tem como pré requisito os seguintes itens:

- PHP na sua versão mais atual;
- MySQL ou MariaDB para banco de dados;
- Apache ou Nginx como servidores de aplicação;
- Suporte ao protocolo HTTPS para melhorias de segurança.

As principais vantagens do WordPress são: solução profissional, código aberto (Open source), atualização frequentes, documentação completa, painel de gerenciamento intuitivo, inúmeros temas e plugins, enfim, apresenta recursos para o desenvolvimento de excelentes soluções.

3.3 Recursos Físicos

Para a implantação da estrutura da aplicação há necessidade de utilização de servidores apropriados para atender a demanda, sendo especificados conforme a demanda do uso e também considerando quantidade de recursos como reserva técnica.

Os equipamentos que irão compor a rede, como switchs e roteadores também serão analisados e planejados conforme a carga de utilização e também sua respectiva reserva técnica. Os elementos estruturais como rack's e nobreak, também precisão ser considerados e avaliados para fornecer segurança e condições térmicas ideais.

Na sequência estão demonstrados a topologia da rede e os respectivos equipamentos que fazem parte. Segue:

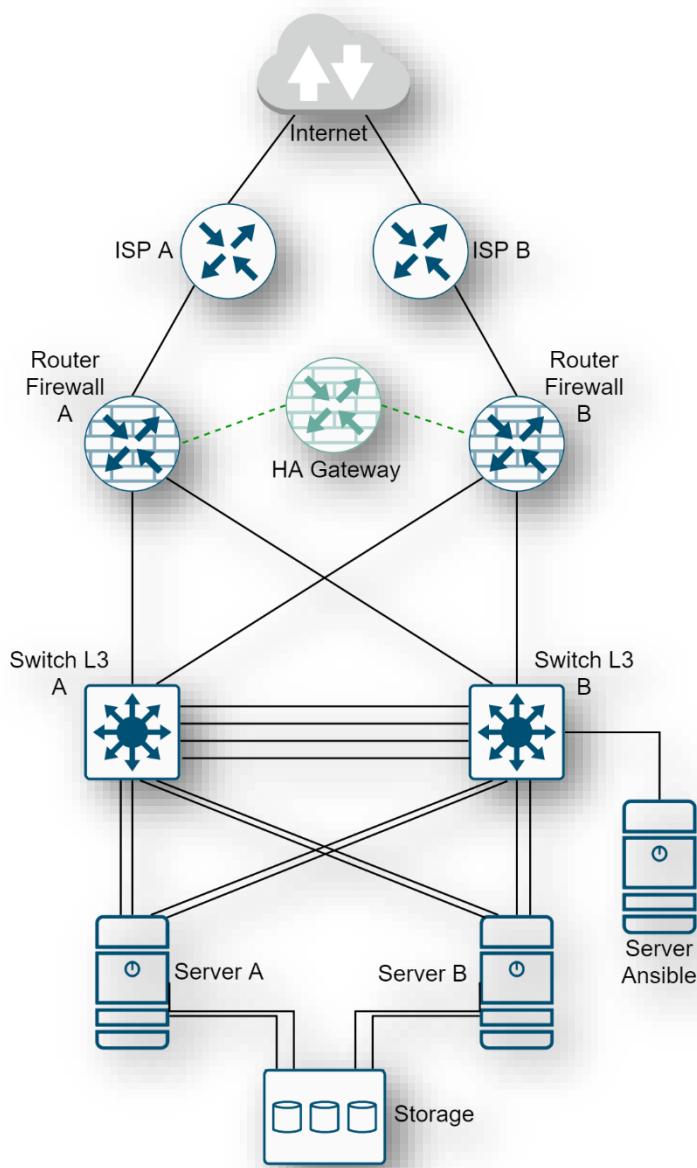


Figura 4 – Topologia Física

Para demonstrar o resultado final da estrutura física é apresentado na imagem abaixo o layout final da montagem do Rack.

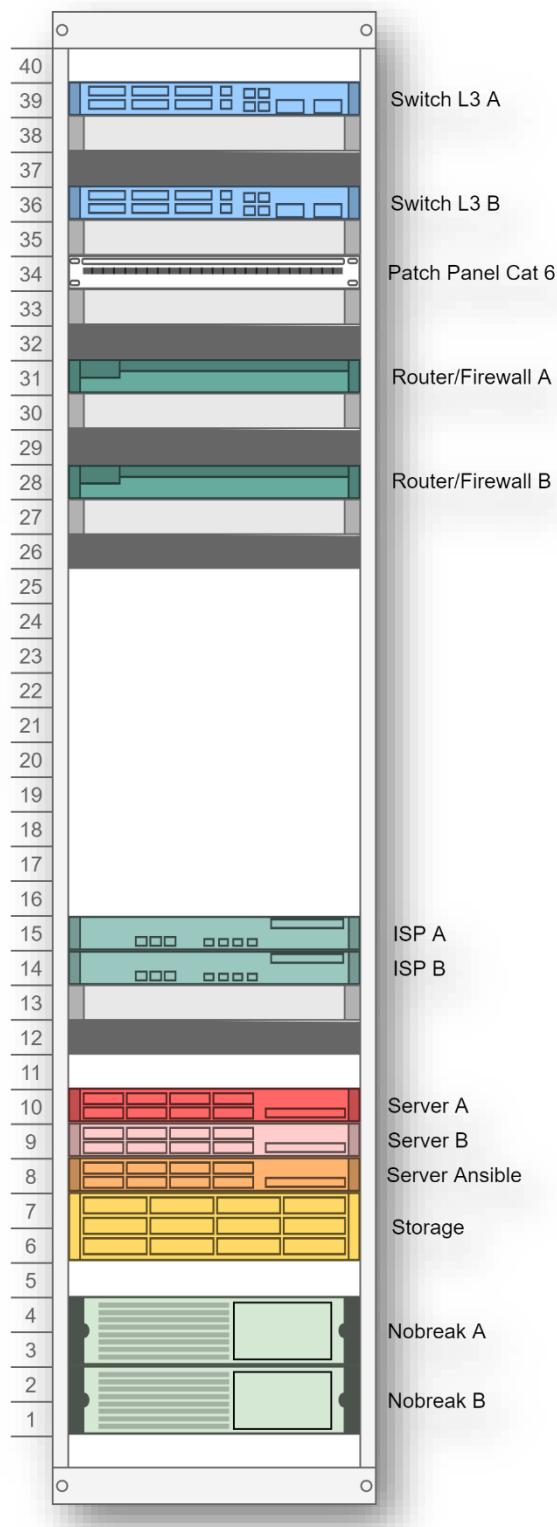


Figura 5 – Montagem Rack da Estrutura de Rede Física

Definidos os componentes físicos é possível desenvolver os demais aspectos e componentes da estrutura.

3.4 Virtualização

As soluções Vmware são fundamentais para a composição da estrutura utilizada para a execução da aplicação e os recursos utilizados. O vSphere 7 será responsável pela virtualização dos hosts e gerenciar as máquinas virtuais, enquanto que o vCenter fará o gerenciamento destes hosts e fornecer recursos de redundância através de soluções adicionais.

Na imagem abaixo está representada a topologia de funcionamento do Vmware, segue:

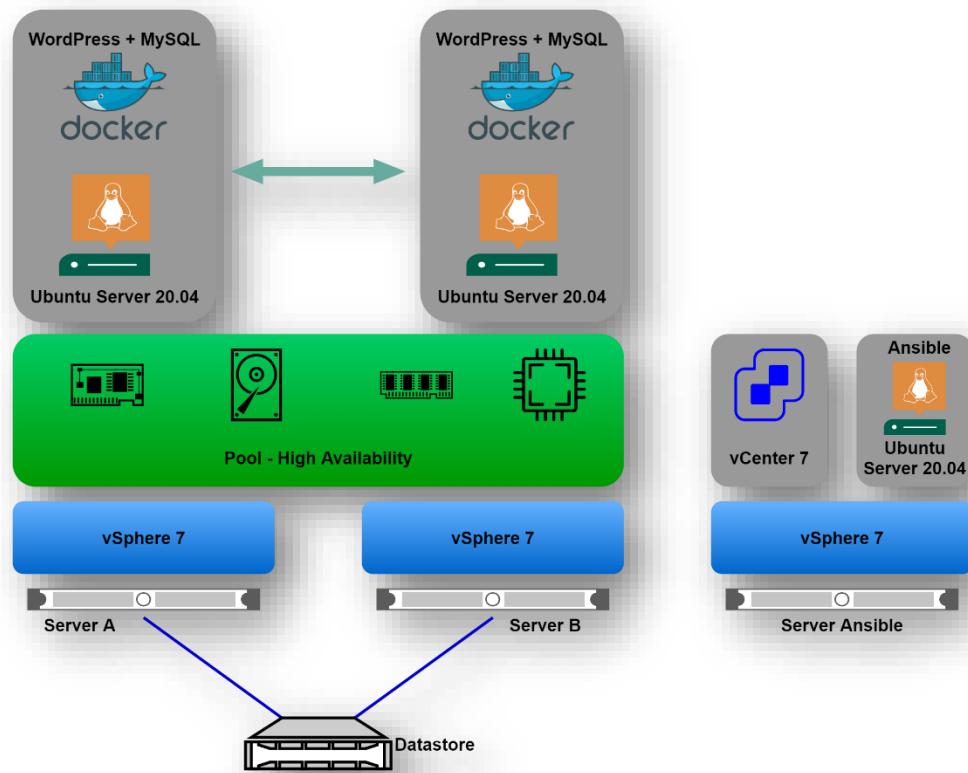


Figura 6 – Estrutura Vmware

Para realizar as funções de redundância é indicado o recurso vSphere HA (High Availability – alta disponibilidade), que consiste num serviço de monitoramento capaz de verificar problemas ou inconsistências. A partir destes alarmes são realizadas tarefas para recuperação do funcionamento do cluster, como religar máquina virtual em outro host.

3.5 Plano de Implementação

O plano de implantação, ou cronograma, será realimentado e atualizado conforme a evolução das etapas e de conhecimentos adquiridos, e com certeza o desenvolvimento desta ferramenta fundamental para qualquer projeto sofrerá correções contantes.

Tarefa	Etapas				
	1	2	3	4	5
Estrutura Rack e Rede					
Compra Servidores e Instalação					
Virtualização e Recursos					
Sistemas Operacionais e Docker					
Código Fonte e Aplicação					
Código Ansible					
Testes Aplicação					
Testes Estrutura					
Pós Configurações					

Tabela 2 – Plano de Implantação

O período necessário para mudança entre cada etapa deve durar 15 dias, sendo que o planejamento total é de 60 dias para implantação, sendo que após este período será necessário realizar configurações e revisões da estrutura, desde a aplicação, código do ansible e até mesmo a rede e servidores.

4 Metodologia do Projeto

O desenvolvimento do projeto é baseado no princípio Devops, onde são utilizados recursos de integração contínua, onde a equipe do desenvolvimento e até mesmo implantação cooperam para que o projeto de implantação seja realizado de maneira eficiente e dinâmico.

O Ansible é utilizado pelo profissional Devops para orquestrar as etapas do projeto, sendo que este código é armazenado numa Git. A Git é uma ferramenta de repositório para

códigos e projetos, que permite a gravação e controle de versões, o download e upload das informações, inclusive possibilita a criação de ramificações de etapas do projeto. O uso desta metodologia permite que os códigos de implantação sejam mantidos em um único local e podendo ser alterados a qualquer momento pelos desenvolvedores.

4.1 Recursos

Para que o projeto possa ser implantado, foi escolhido o Ansible para a orquestração das implantações do virtualizador, máquinas virtuais, sistema operacional, aplicações e configurações, sendo o seu código armazenado através do acesso ao portal Git.

O virtualizador utilizado no projeto é o Vmware vSphere na sua versão 7 e do tipo Standard, a qual possui em seu pacote os seguintes recursos principais: vCenter Server Standard, vMotion, vShield Endpoint, Replication, entre outros.

As máquinas virtuais que irão operar o Ansible e a aplicação Docker, terão como sistema operacional o Linux Ubuntu Server na versão 20.04 e 64 bits. Este sistema operacional possui compatibilidade e vasta documentação na internet, o que oferece segurança e credibilidade para operação de ambos recursos.

O vCenter Server na versão 7, é instalado para ser executado em uma máquina virtual pré configurada, utilizando o sistema operacional VMware Photon Linux na sua versão 1.0. Esta aplicação será implantada no servidor Ansible, oferecendo o gerenciamento dos três servidores e criando a alta disponibilidade entre os servidores A e B.

No servidor A será criado a máquina virtual com o sistema operacional Ubuntu Server e dentro deste, a aplicação Docker. Neste gerenciador serão criados os dois containers necessários para executar a aplicação: WordPress e MySQL. Por questões de segurança estes dois containers serão separados, permitindo criar uma rede isolada para que somente estes dois módulos se comuniquem, não permitindo acesso do ambiente exterior ao banco de dados.



Figura 7 – Diagrama de Integração e Orquestração

Conforme a imagem acima demonstra, é possível verificar a troca de informações entre os módulos que fazem parte deste projeto.

4.2 Considerações

Antes de demonstrar os aspectos que foram utilizados para a seleção dos recursos citados acima, é preciso identificar o motivo pela escolha de utilizar uma estrutura física implantada na própria empresa.

A escolha de cada aplicação, recurso e sistema operacional é baseada em aspectos técnicos, financeiros e de credibilidade. O virtualizador da VMware foi selecionado devido aos recursos técnicos e a confiabilidade oferecida, principalmente em consideração a utilização na estrutura atual da empresa, já que mesmo utilizando a versão ESXi na modalidade gratuidade é notória a confiabilidade do produto. Este investimento é proporcional a segurança que trará para o funcionamento da estrutura.

A opção de nuvem pública não apresenta vantagem por demandar pagamento de valor mensal, principalmente pela razão de a empresa já possuir uma estrutura mínima já montada, precisando de investimentos pontuais e que não seriam altos comparados ao valor da nuvem pública.

O sistema Linux Ubuntu na versão Server oferece segurança e material de suporte, que aliado a gratuidade da licença torna a solução ideal e compatível. Também é preciso ser considerado o custo e o risco relacionado ao suporte para instalação e manutenção, mas este valor é justificado comparado ao valor de licenças de Windows Server, por exemplo.

O gerenciador de conteúdos WordPress também apresenta a mesma situação do sistema operacional Ubuntu, ou seja, não há valor de licenciamento e a ferramenta é amplamente utilizada apresentando confiabilidade e acesso a documentação.

A escolha do Docker foi tomada considerando a agilidade em implantação e gerenciamento. Poder implantar serviços específicos sem demandar instalações completas de sistema operacional também foi levada e consideração.

5 Arquitetura

A aplicação objeto deste projeto é baseada no WordPress, fornecendo um website com recursos específicos para as características da empresa Digicomp. Na imagem abaixo é possível visualizar as camadas que compõem a aplicação.



Figura 8 – Arquitetura da Aplicação

O modelo é composto por três camadas: Apresentação, Aplicação e Banco de Dados. As duas primeiras estão sendo concebidas para executarem no mesmo container, portanto a troca de informações entre ambas é direta e sem necessidade de conexões externas.

Já o container com o banco de dados estará isolado conectado ao container do WordPress através de rede virtual utilizando VLan separada e isolada, principalmente da rede externa e internet.

Mesmo o portal estando preparado para fornecer o acesso externo é necessário criar regras de segurança, feito através da utilização de Firewall de rede, a qual garantirá acesso da internet a somente portas e protocolos configurados.

6 Requisitos Técnicos

Após abordar o detalhamento do software é necessário relacionar os equipamentos necessários para o funcionamento. Como visto anteriormente os servidores são os elementos centrais da estrutura, necessitando de configurações compatíveis com a carga de processamento e armazenamento.

Os Servidores Ansible, A e B utilizados neste projeto consistem no Lenovo ThinkSystem SR630 Server com as seguintes características:

- 02 processadores Xeon Silver 4214 12 core 24 threads 2.2GHz
- Memória RAM de 128 GB DDR4 2933MHz ECC Reg.
- Placa controladora raid 7308i 2GB cache (raid 0, 1, 5, 10, 50 ou 60)
- 2 fontes de 750w Hot Plug
- 08 interfaces de rede gigabit, sendo 04 on board e 04 conexões através de placa de rede ThinkSystem 1GB 2-port RJ45
- 03 Discos hot-swap 2.5" SSDS NVMe 500GB em Raid 5
- Gabinete Rack 1U



Figura 9 – Servidor Lenovo ThinkSystem SR630

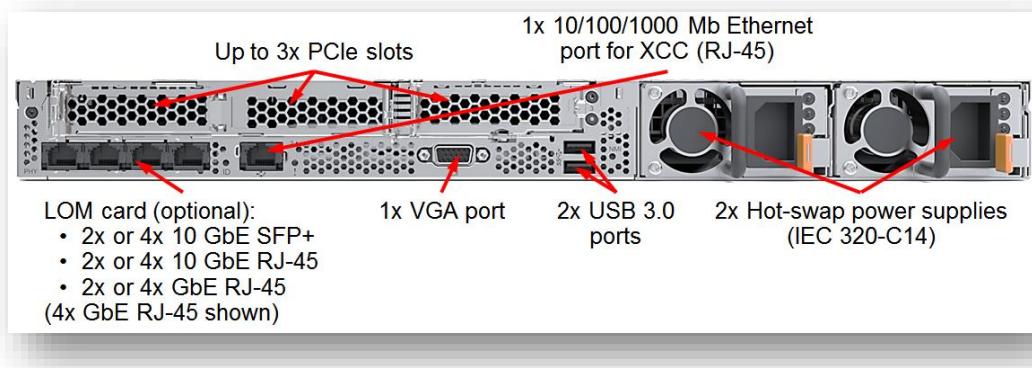


Figura 10 – Servidor Lenovo ThinkSystem SR630 – Vista conexões traseiras



Figura 11 – Servidor Lenovo ThinkSystem SR630 – Layout armazenamento

O Storage utilizado é do fabricante Lenovo, modelo Thinksystem Ds2200 com as seguintes especificações:

- 8 discos SSD de 4TB enterprise
- 2 interfaces 10GB e 2 intefaces 2.5GB
- Fonte de alimentação 250W

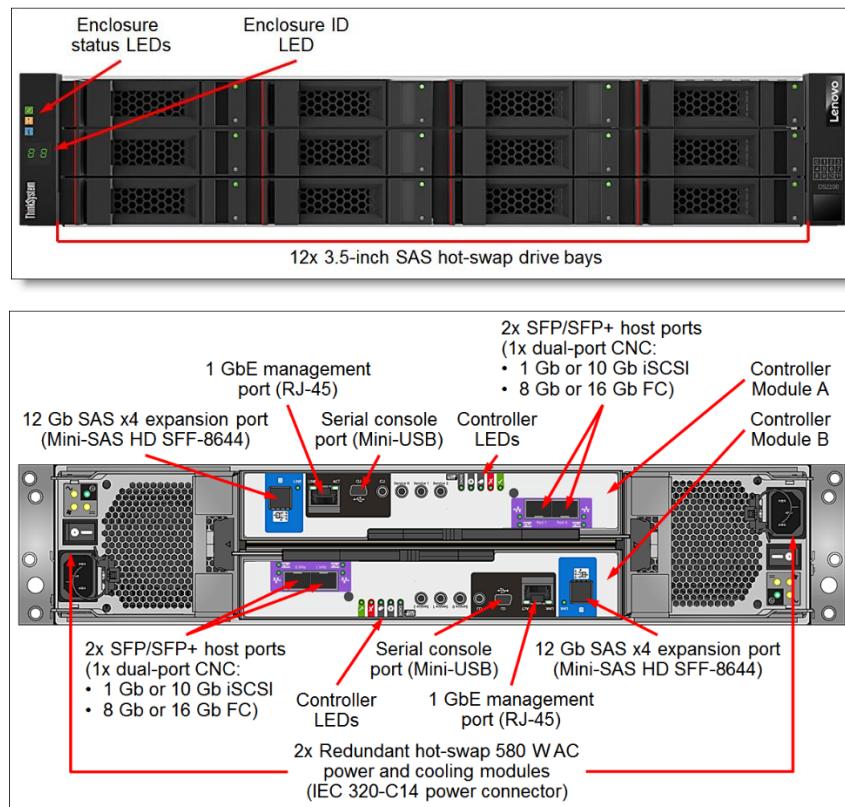


Figura 12 – Armazenamento

Os switchs utilizados neste projeto são o Dell modelo SG 5204 MR, possuindo as seguintes características técnicas:

- 24 portas de rede gigabit, 02 portas SFP de 10GbE e 2 portas 1GbE
- Layer 3
- Empilhável
- Memória Flash 1 GB
- Capacidade da malha 212 Gbps
- Buffer pacote 4MB



Figura 13 – Switch Dell NC3024F

O roteador C1905 da Cisco tem a função de Firewall e roteador, podendo gerenciar a rede e assegurar segurança. Segue especificações:

- 2x GbE RJ45 10/100/1000
- Memória RAM 256MB DDR2
- Memória Flash 256MB
- Módulo switch integrado 4 portas 10/100/1000



Figura 14 – Netgate SG5100

Na sequência é demonstrado tabela contendo o levantamento dos custos dos equipamentos. Segue:

Equipamento	Descrição	Qtde	Valor Unit. (R\$)	Valor Total (R\$)
Servidor	Servidor Lenovo SR650	03	45.000,00	135.000,00
Storage	Storage Lenovo DS2200 com 8 discos de 4TB	01	65.000,00	65.000,00
Switch	Switch Layer 3 Dell N3024F	02	10.500,00	21.000,00
Router	Roteador Cisco C1905 com Módulo adicional 4 portas 1Gb	02	14.500,00	29.000,00
TOTAL R\$				250.000,00

Tabela 3 – Relação de equipamentos

Além destes valores de equipamentos também é preciso ser considerado os valores de implantação dos sistemas operacionais. A Licença do VMware vSphere 7 na versão Standard precisa ser adquirida, juntamente com o serviço especializado de implantação.

Mesmo utilizando sistemas operacionais sem custo para licenciamento, como o Ubuntu, é necessário considerar valores para implantação dos mesmos ou então treinamento da equipe técnica. Segue tabela com valores de implantação e mensais de manutenção:

Item	Descrição	Qtde horas	Valor Unit. (R\$)	Valor Total (R\$)
01	Implantação VMware vSphere 7 Standard, vCenter 7 e HÁ	10	160,00	1.600,00
02	Sistemas Operacionais Ubuntu	06	160,00	960,00
03	Suporte Mensal Sistemas Operacionais Ubuntu	01	1.500,00	1.500,00
04	Desenvolvimento Site WordPress	40	160,00	6.400,00
05	Suporte Mensal manutenção site WordPress	01	2.500,00	2.500,00
06	Montagem estrutura de servidores e equipamentos de rede (roteadores e switchs)	20	160,00	3.200,00
TOTAL IMPLANTAÇÃO				12.160,00
TOTAL SUPORTE MENSAL				4.000,00

Tabela 4 – Custos de implantação e suporte mensal

Com base nas informações obtidas é possível avaliar o desenvolvimento do projeto em sua capacidade idealizada para atender a demanda de clientes no acesso ao portal da empresa.

7 Implantação

Após desenvolver o conceito do projeto ideal, cabe apresentar o ambiente em que foram realizados os testes e simulações de uma estrutura básica, mas com os componentes fundamentais. Os próximos tópicos apresentaram de maneira objetiva e focada na simulação, os aspectos vistos anteriormente.

7.1 Modelo de Simulação

Para realização dos testes da implantação foi utilizada topologia simplificada, mas que conta com recursos e componentes fundamentais para a análise de desempenho. Segue imagem com a demonstração.

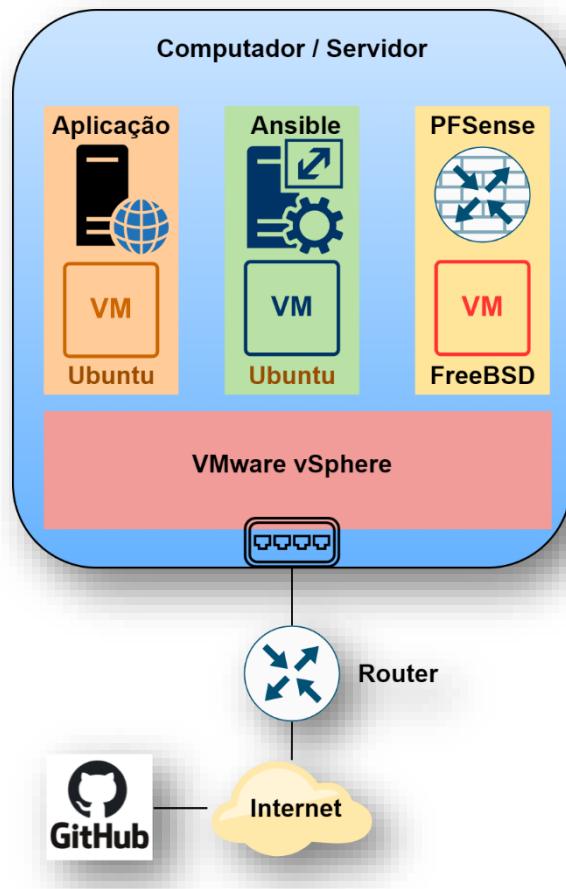


Figura 15 – Topologia Básica

O virtualizador utilizado e o sistema operacional das máquinas virtuais do Ansible e da aplicação também serão Ubuntu, tendo como diferença somente versões mais simplificadas e com menos recursos.

7.2 Arquitetura

O modelo da arquitetura utilizado nesta simulação segue o princípio do tópico descrito anteriormente neste documento. O WordPress será a plataforma de gerenciamento do portal da empresa Digicomp, sendo este composto pelos recursos Apache, PHP e MySQL.

Estes componentes estarão sendo implantados através de containers no Docker, sendo toda esta estrutura executada no sistema operacional Linux Ubuntu. A implantação destes será realizada em parte de maneira manual, respeitando características de cada instalação, e em parte de maneira automatizada através do Ansible.

7.3 Metodologia

O foco deste projeto é demonstrar a execução de aplicação utilizando recursos de orquestração e automação. O Ansible será o agente responsável por estas ações, tendo o seu código de execução armazenado em um portal Git, a qual será utilizado como fonte da ação.

A ideia utilizada no laboratório também vai de encontro ao mencionado no item referente a topologia em seu modelo ideal de implementação. A capacidade e os recursos utilizados em menor quantidade não irão retirar a capacidade de demonstrar o objetivo.

7.4 Hardware

O computador utilizado como servidor é equipado com as seguintes especificações:

- Processador E3-1271 v3 4 Core
- Memória RAM 32 GB
- 03 Unidades: SSD de 120GB e 02 SHDD de 1TB cada
- Placa mãe Asus Z97M Plus
- Placa de Rede Mikrotik Gigabit
- Fonte 500W
- Gabinete 2 baias

O Computador é conectado ao Switch TPLINK de 16 portas Gigabit, modelo SG-1016D. O roteador ligado ao GPON da operadora de internet é um Mikrotik RouterBoard RB 750Gr3.

7.5 Topologia

A topologia utilizada para a implantação da estrutura é demonstrada abaixo através da imagem, segue:

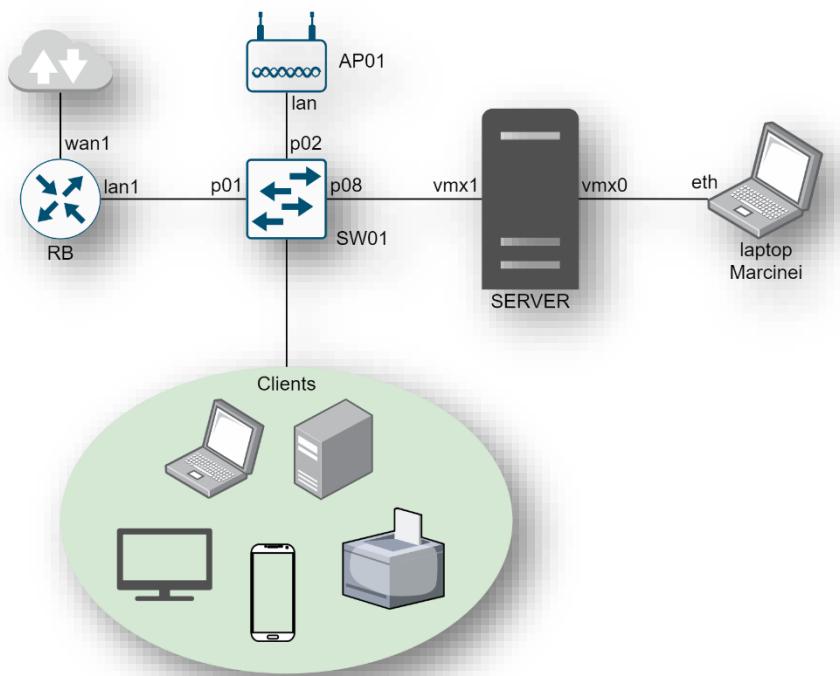


Figura 16 – Topologia Modelo Simulação

Conforme a imagem é possível verificar a existência da rede principal e a conexão do servidor a esta. O Laptop está conectado diretamente a porta de rede específica da rede pertencente a estrutura do projeto.

Importante demonstrar também o endereçamento das redes e dos respectivos equipamentos e sistemas operacionais. Segue tabela:

Equipamento	Port	IP/Máscara	Gateway
RB	lan1	10.0.2.1/24	
	wan	191.37.245.47	10.101.101.3
Server	vmx1	10.0.2.39/24	10.0.2.1
	vmx2	10.0.4.1/24	10.0.4.1
VM PFsense	lan	10.0.4.1/24	10.0.4.1
VM Ansible	lan	10.0.4.4/24	10.0.4.1
VM Freenas	lan	10.0.4.5/24	10.0.4.1
VM App	lan	10.0.4.6/24	10.0.4.1

Tabela 5 – Endereçamento IP

Com estas informações é possível identificar os equipamentos e suas respectivas conexões de rede.

7.6 Virtualização

Se faz necessário demonstrar como está montado a topologia da virtualização do servidor, suas interfaces de rede, processamento, memória e armazenamento. Segue imagem representando:

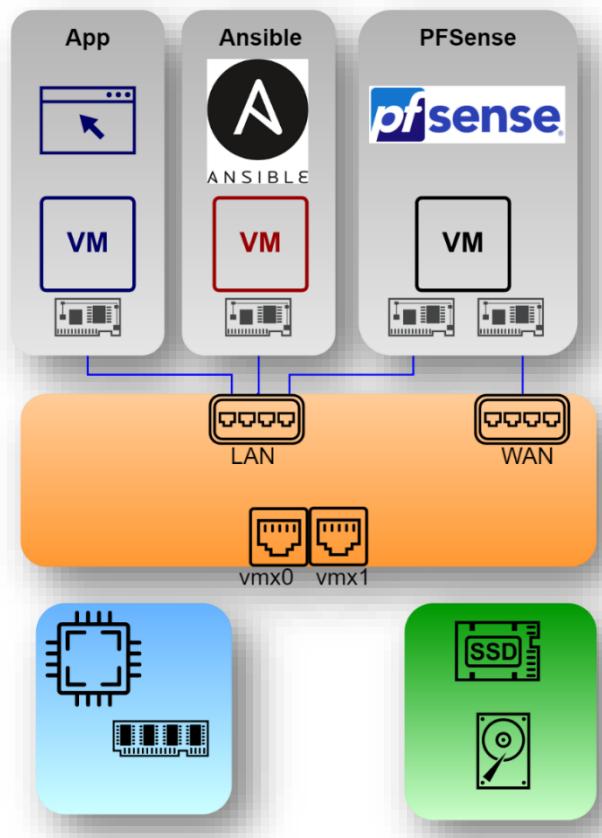


Figura 17 – Topologia Virtualização

Abaixo estão relacionados os detalhes das máquinas virtuais utilizadas na estrutura:

VM	Descrição
PFSense	<ul style="list-style-type: none"> • 01 vCore • 1GB • 20 GB (datastore Laboratorio) • Interfaces de Rede • SO: FreeBSD 12 x64
App	<ul style="list-style-type: none"> • 02 vCores • 4GB • 20 GB (datastoreA – FreeNas) • 1 Interface de Rede • SO: Ubuntu 20.04 x64

Ansible	<ul style="list-style-type: none">• 02 vCores• 4GB• 50GB (datastore Laboratio)• 1 Interface de Rede• SO: Ubuntu 20.04 x64
----------------	---

Tabela 6 – Relação de Máquinas Virtuais

Após relacionar as informações necessárias da estrutura, segue os procedimentos de implantação e instalação.

7.7 Plano de implantação

O plano de implantação consite nas tarefas a serem executadas para que a aplicação seja instalada e operacional. Abaixo está disponibilizada imagem com as etapas. Segue:

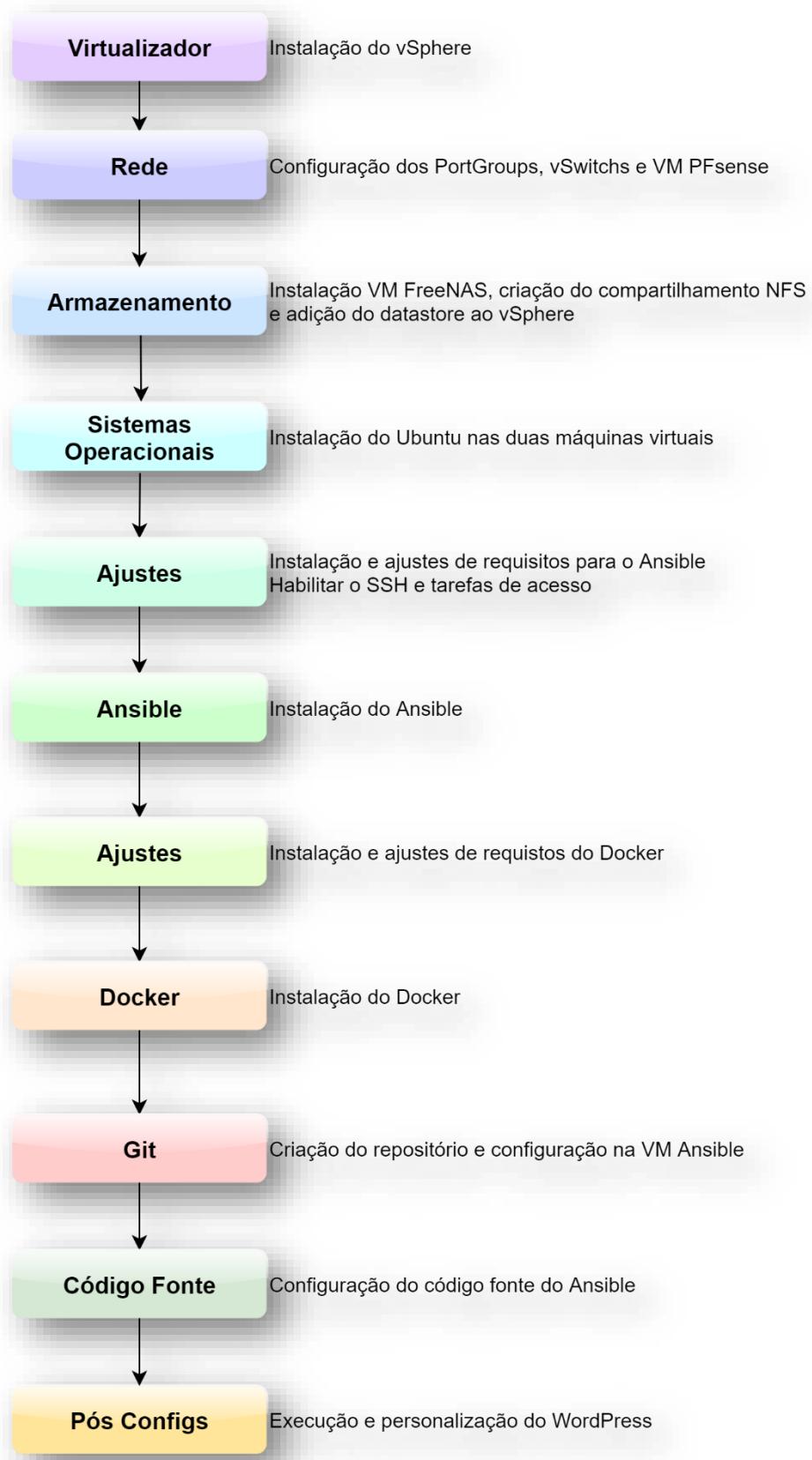


Figura 18 – Plano de Implementação

Com a definição das tarefas é possível demonstrar cada etapa com suas respectivas informações e procedimentos a serem realizados.

7.7.1 Virtualizador

Como a primeira etapa para implantação, é necessário instalar no servidor a solução de virtualização VMware vSphere 7 Standard. A primeira etapa consiste em realizar a gravação da imagem, devendo ser obtido no site de download de produtos da vmware. Segue endereço abaixo:

https://customerconnect.vmware.com/en/downloads/info/slug/datacenter_cloud_infrastructure/vmware_vsphere/7_0

Neste portal será necessário o registro do usuário a qual fez aquisição da licença do produto vmware. Após realizar o download da imagem e realizar a gravação da imagem em pen drive conforme inúmeros tutoriais existentes na internet (rufus por exemplo), será necessário carregar o boot do computador/servidor pelo pen drive. Na tela abaixo está representada a primeira tela do instalador, onde é exibido informações iniciais da aplicação e solicitando a confirmação com a tecla “Enter”.



Figura 19 – Instalador do ESXi 6.7

A tela seguinte solicita a confirmação do aceite do licenciamento do software. Segue:

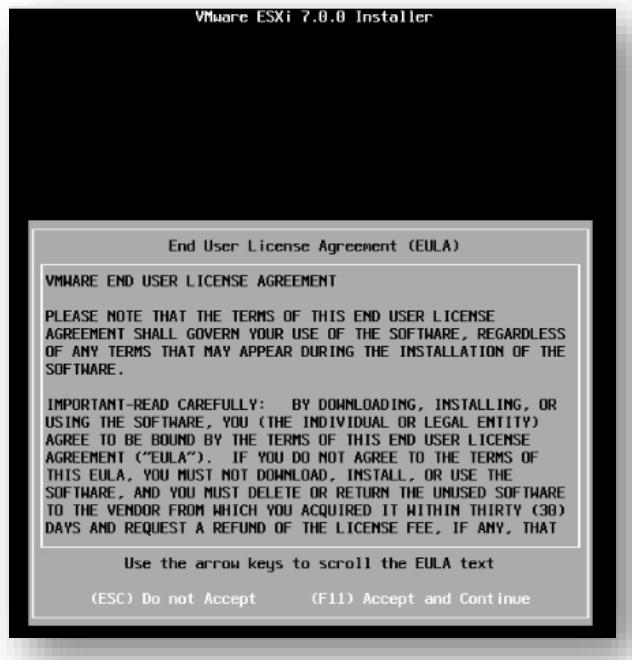


Figura 20 – Aceite do contrato de licenciamento

O próximo passo é indicar o disco que irá armazenar o ESXi, segue imagem:



Figura 21 – Seleção do armazenamento da instalação

O próximo passo é definir o layout do teclado a ser configurado, segue:



Figura 22 – Seleção do layout do teclado

A próxima tela solicita o cadastro da senha do usuário root, segue:



Figura 23 – cadastro da senha do usuário “root”

Segue tela solicitando a confirmação das informações que serão instaladas após pressionar a tecla “F11”.



Figura 24 – Confirmação da instalação

Após concluir o processo de instalação será solicitado a remoção do disco de instalação e em seguida o reboot do servidor. Segue tela:

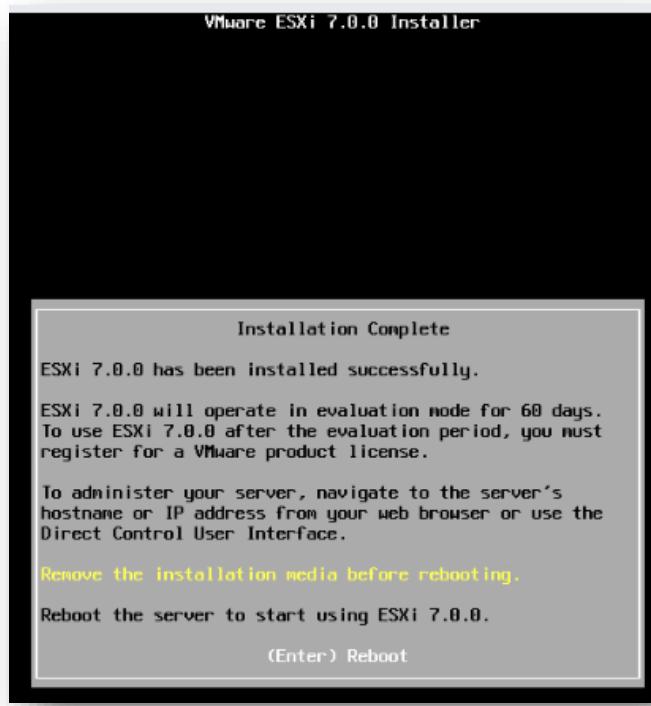


Figura 25 – finalização da instalação

Após o servidor reiniciar com o boot do ESXi é apresentado tela contendo informações do equipamento e endereços para o acesso do site:

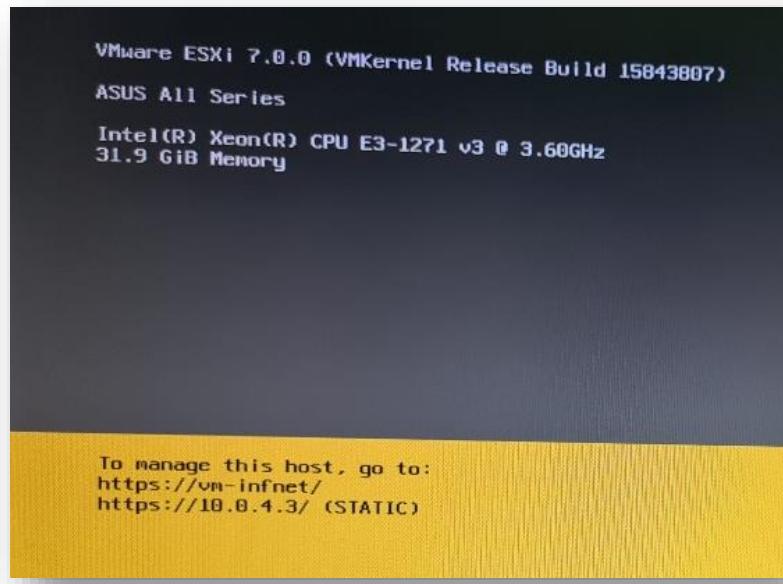


Figura 26 – Tela inicial do ESXi

Para acessar os menus de configuração deve ser pressionado a tecla “F2” e em seguida inserir as informações de login do usuário “root”, segue tela de login:

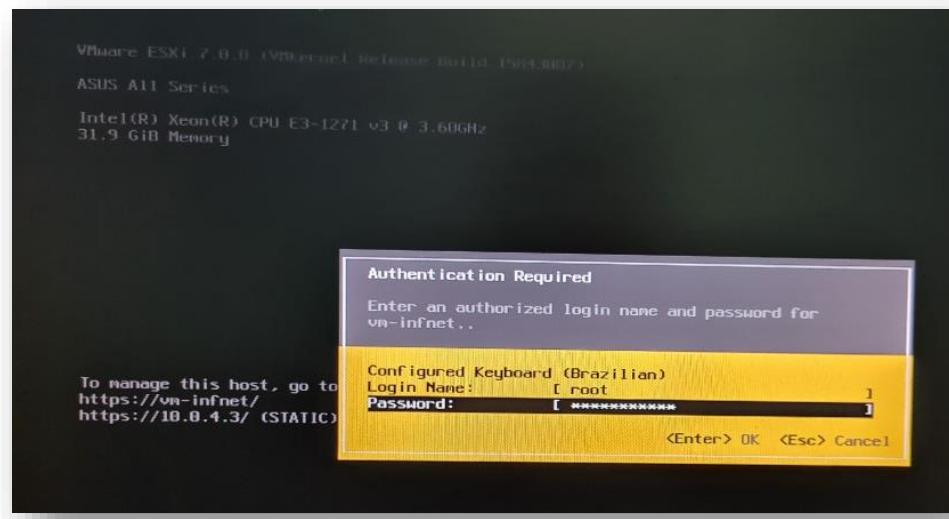


Figura 27 – Tela de login para os menus de configuração

Em seguida estão representadas telas de configuração com as informações referentes a cada opção do menu:

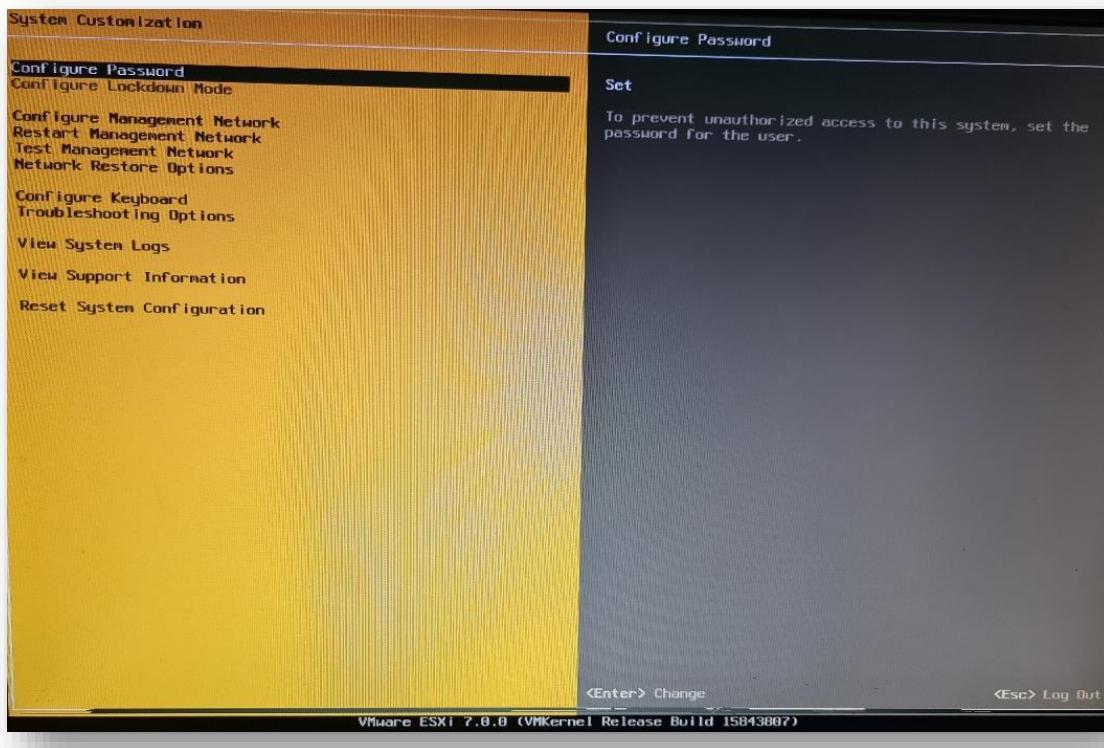


Figura 28 – menu principal

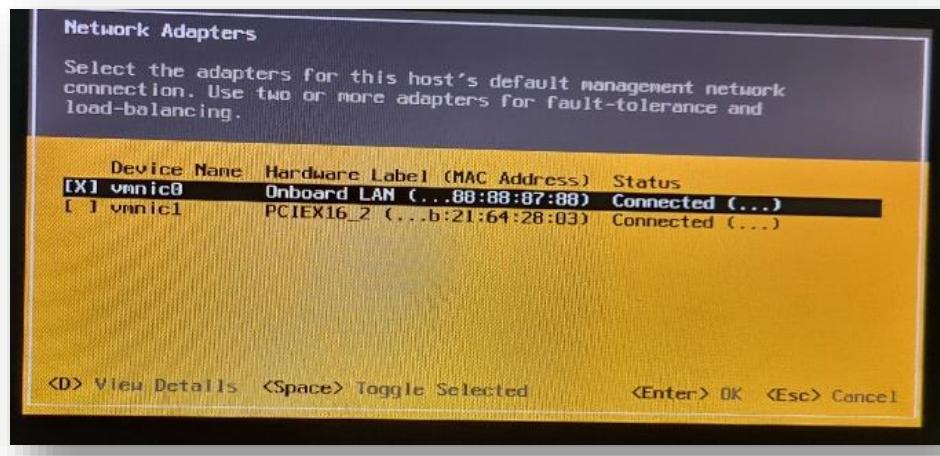


Figura 29 – Placa(s) de rede disponível(eis) e seleção da interface de rede que será definida na Port Group de Gerenciamento

Nas telas seguintes estão representadas a configuração do endereçamento IP do host ESXi, lembrando que é altamente indicado que estas configurações sejam definidas de modo

estático, não tendo possibilidade de perda de conexão ou utilização de endereço diferente do configurado para gerenciamento (vCenter como exemplo). Seguem imagens:

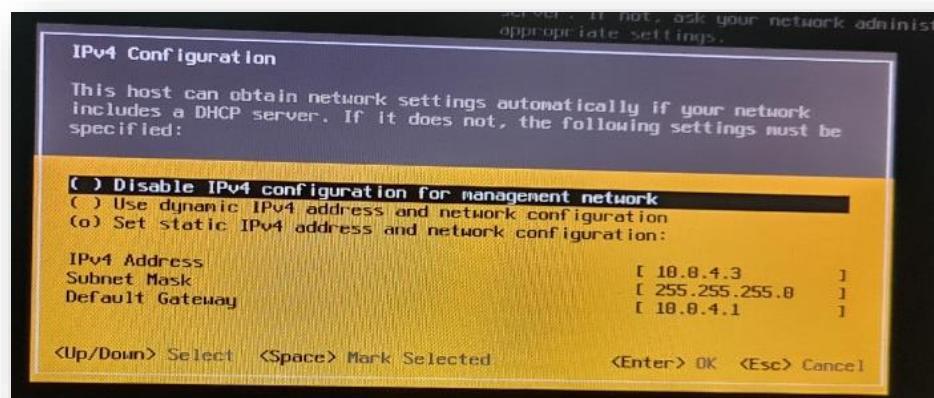


Figura 30 – configuração de endereçamento IP do host

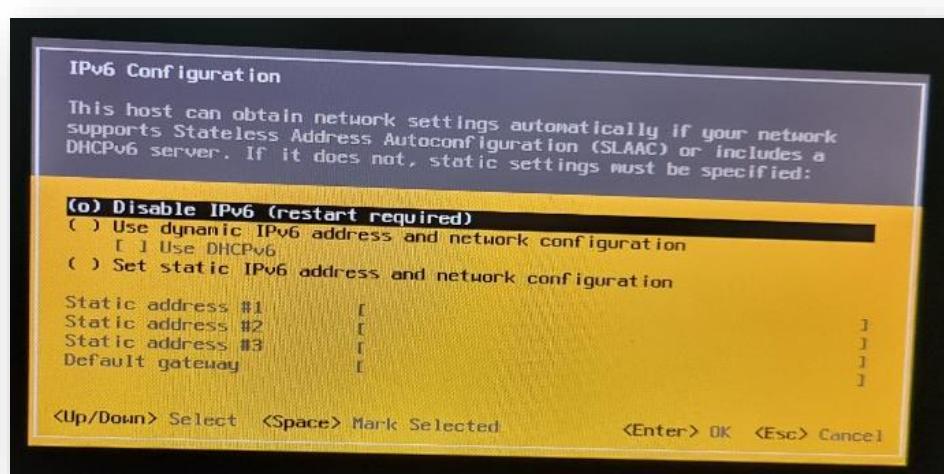


Figura 31 – configuração IPv6, sendo desabilitada para o presente laboratório

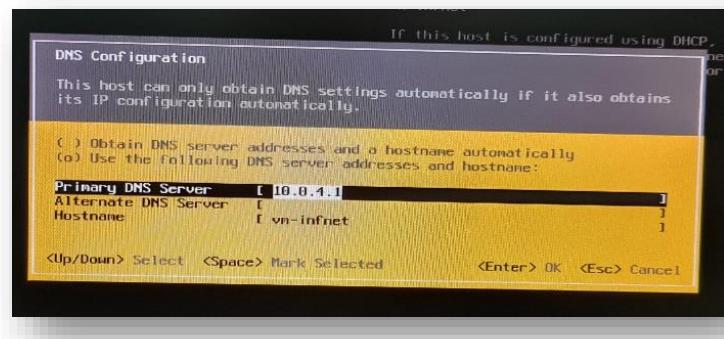


Figura 32 – configurações DNS do host ESXi



Figura 33 – configuração do sufixo dns customizado

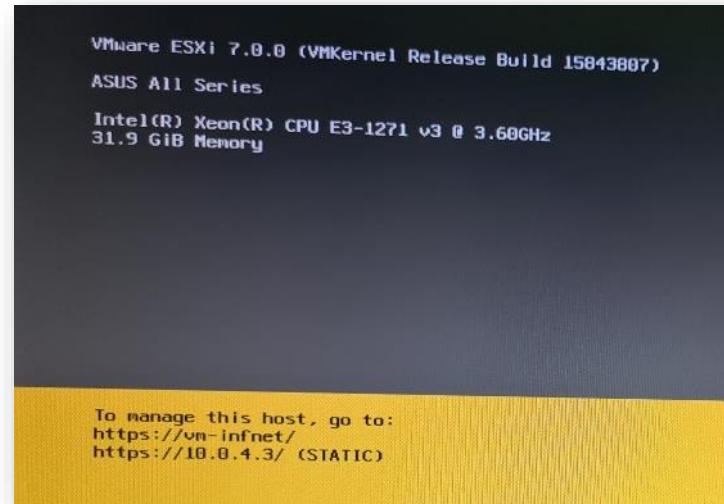


Figura 34 – tela de login com as informações de rede alteradas

Estas informações da rede também podem ser alteradas pelo site do host ESXi, mas para a configuração ideal é que seja realizado via acesso direto ao servidor. Abaixo está demonstrada tela de login ao inserir em qualquer navegador web do endereço IP, segue:

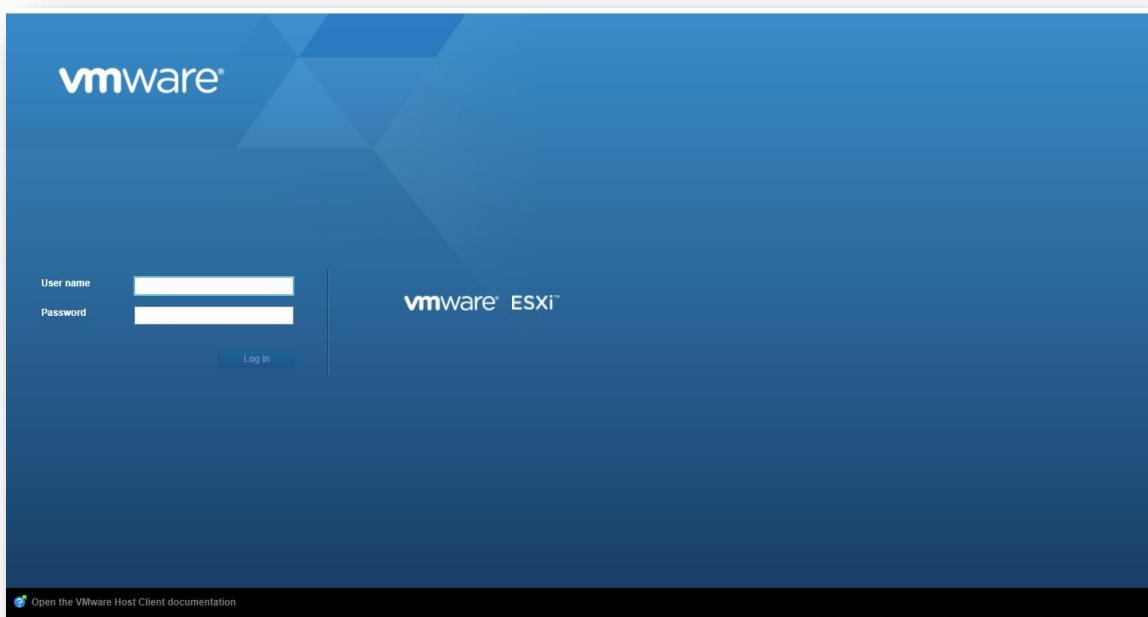


Figura 35 – tela de login do site do host ESXi

Abaixo está capturada tela do portal com as opções e informações necessárias para realizar as configurações das máquinas virtuais, armazenamento e redes. Segue:

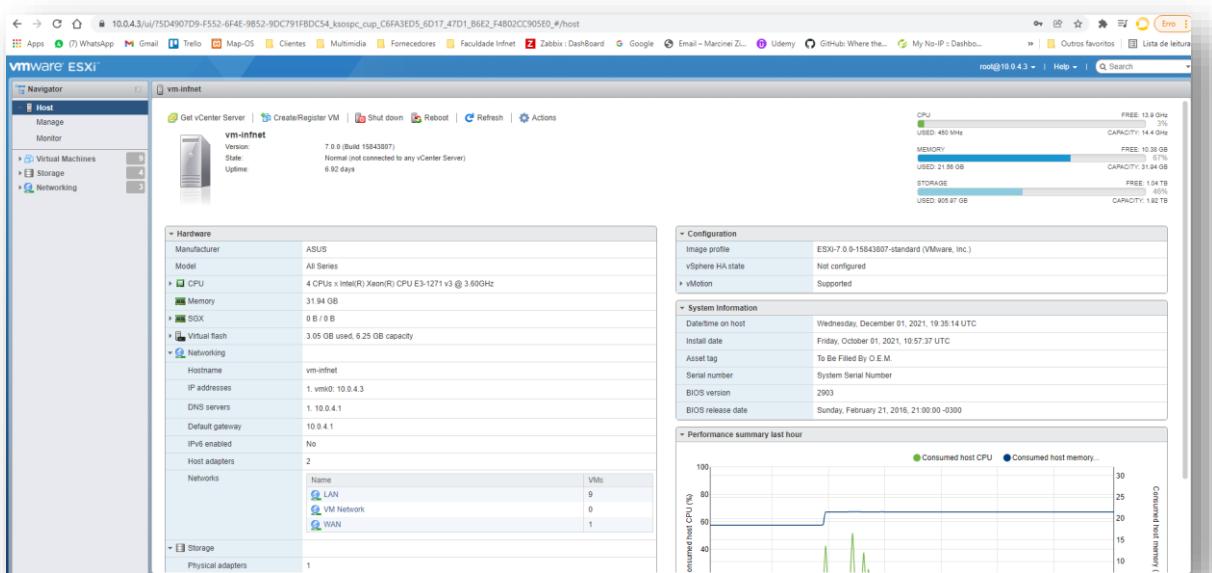


Figura 36 – Dashboard do host ESXi

Finalizada a instalação do virtualizador será necessário configurar as questões relacionadas a rede.

7.7.2 Rede

A primeira etapa da configuração da rede no host VMware é verificar a existência das duas placas de rede instaladas no servidor. Segue imagem demonstrando as interfaces.

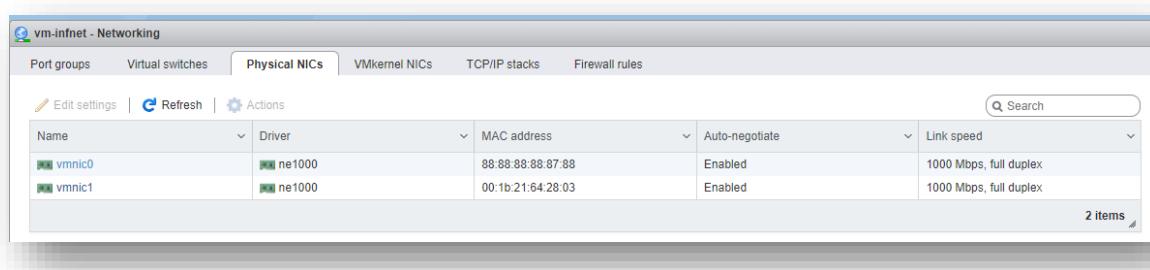


Figura 37 – Interfaces de rede físicas

Em seguida será necessário criar os switchs virtuais, onde depois serão adicionadas as Port Groups e por consequência os apontamentos das máquinas virtuais. Segue imagem demonstrando as duas vSwitch's criadas.

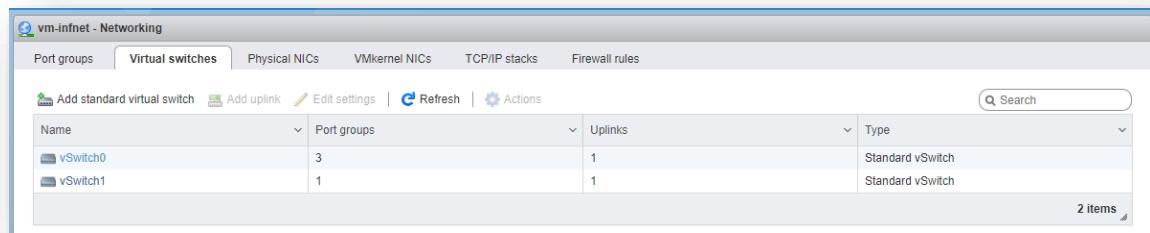


Figura 38 – Lista das vSwitch's

A vSwitch0 está configurada para conectar a interface de rede vmnic0, enquanto que a vSwitch1 aponta para a NIC vmnic1. Nas imagens abaixo estão representadas as configurações finalizadas dos respectivos switchs virtuais. Segue:

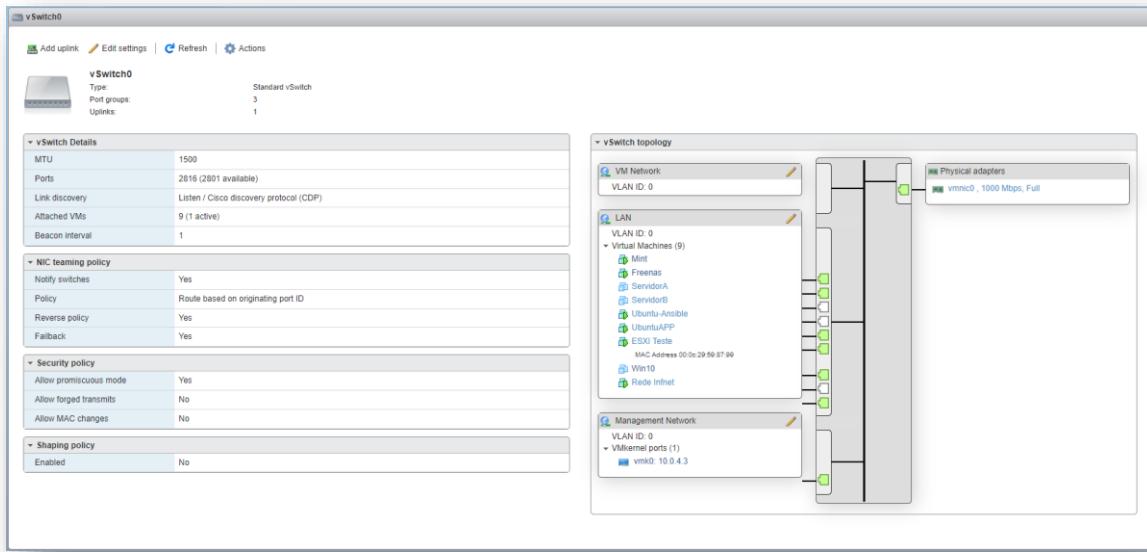


Figura 39 – Configuração vSwitch0

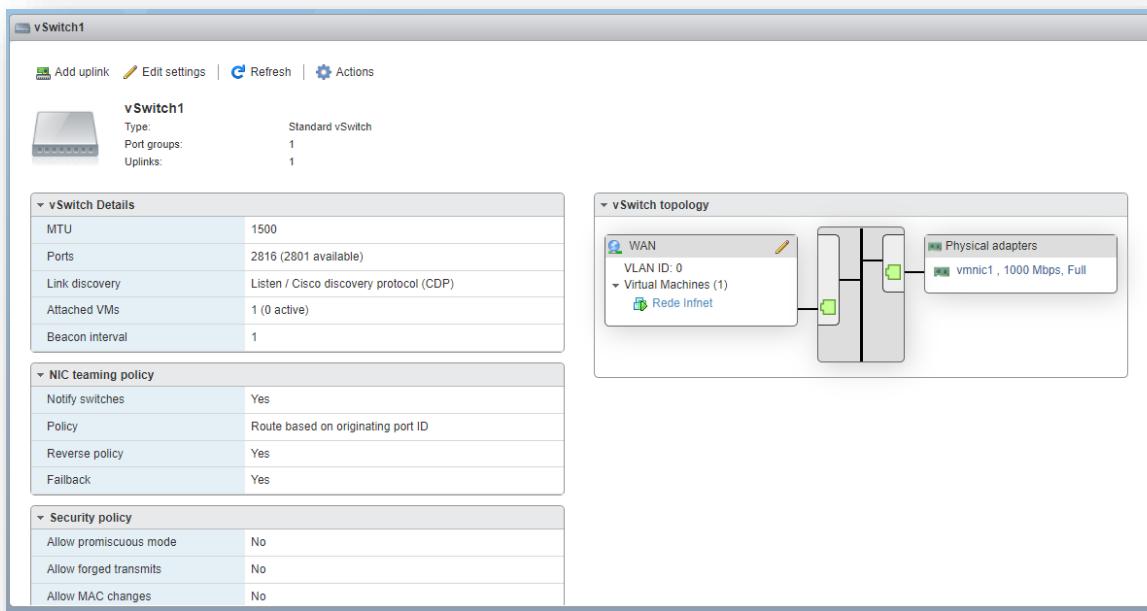


Figura 40 – Configuração vSwitch1

Desta forma é possível iniciar a configuração dos Port Groups, sendo criados os seguintes: LAN (para a rede interna do virtualizador) e WAN (conectada diretamente a rede com internet). Segue imagem com a lista das portas criadas.

Name	Active ports	VLAN ID	Type	vSwitch	VMs
VM Network	0	0	Standard port group	vSwitch0	0
LAN	6	0	Standard port group	vSwitch0	9
Management Network	1	0	Standard port group	vSwitch0	N/A
WAN	1	0	Standard port group	vSwitch1	1

Figura 41 – Lista de Port Groups

Na sequência serão demonstradas as portas LAN e WAN, com as suas respectivas conexões e topologia. Segue:

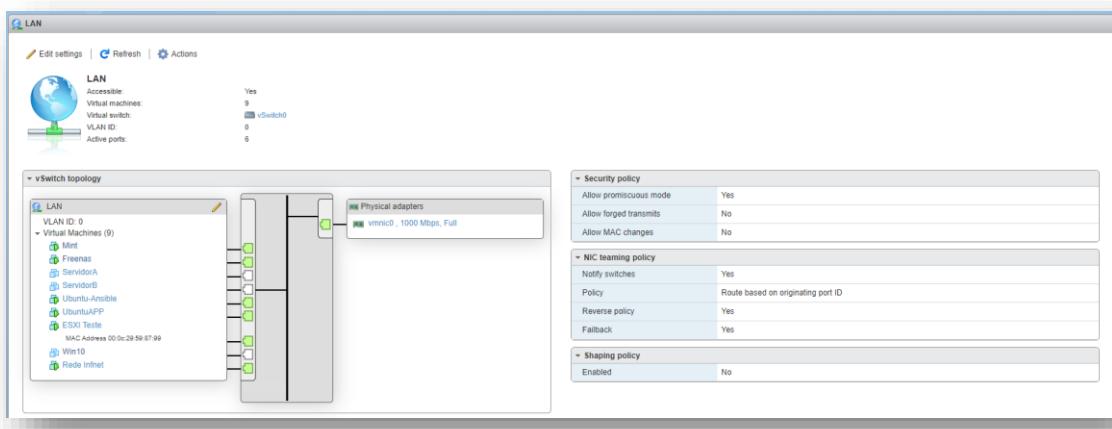


Figura 42 – Port Group LAN

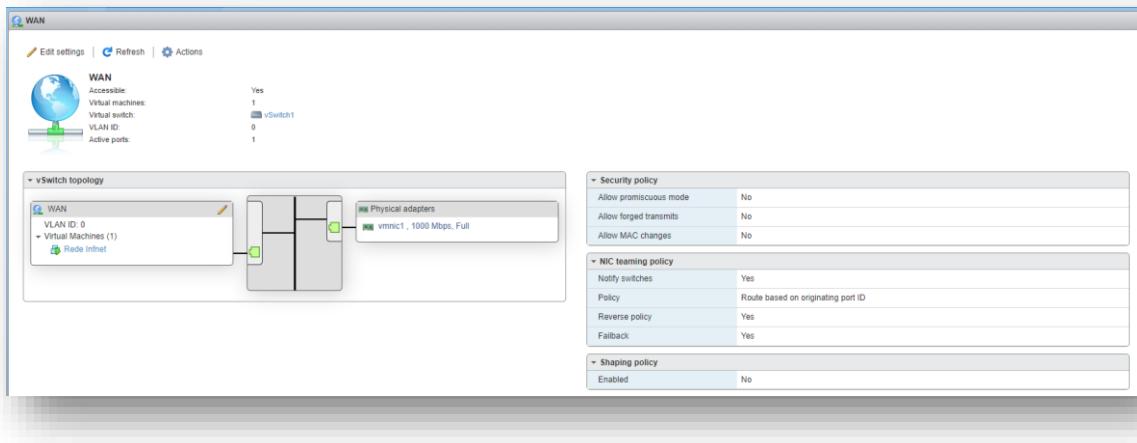


Figura 43 – Port Group WAN

Para simular o funcionamento da rede de maneira completa foi criado um roteador e firewall, onde a solução utilizada foi o projeto opensource PFsense, baseado em FreeBSD 12 x64. A máquina virtual criada para receber o PFsense está representada abaixo:

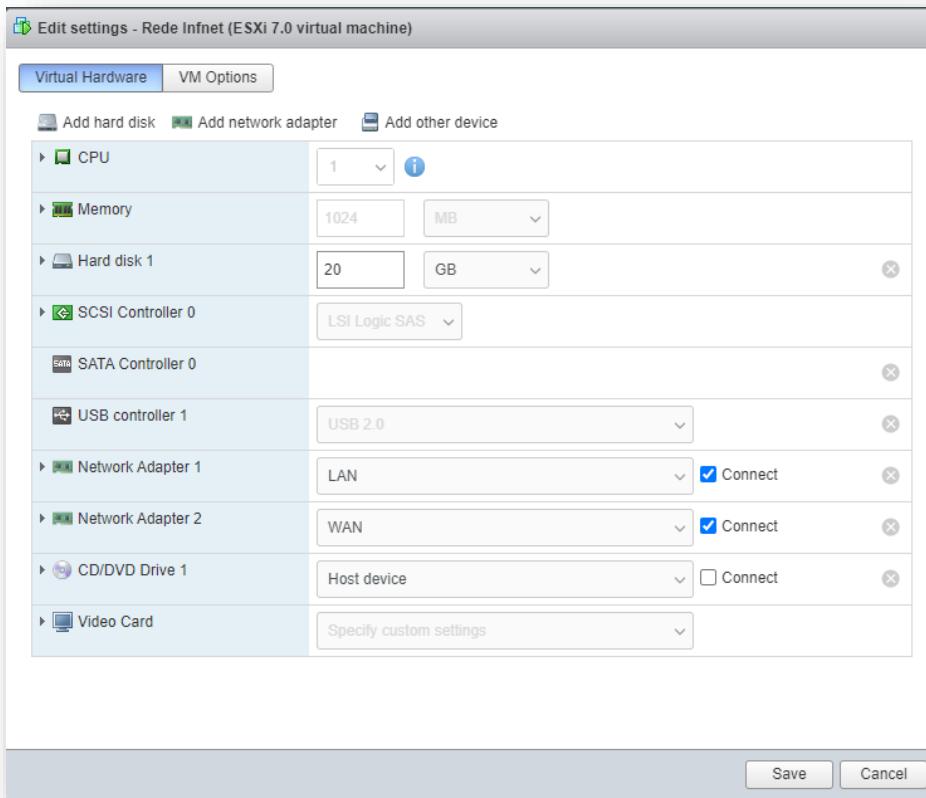
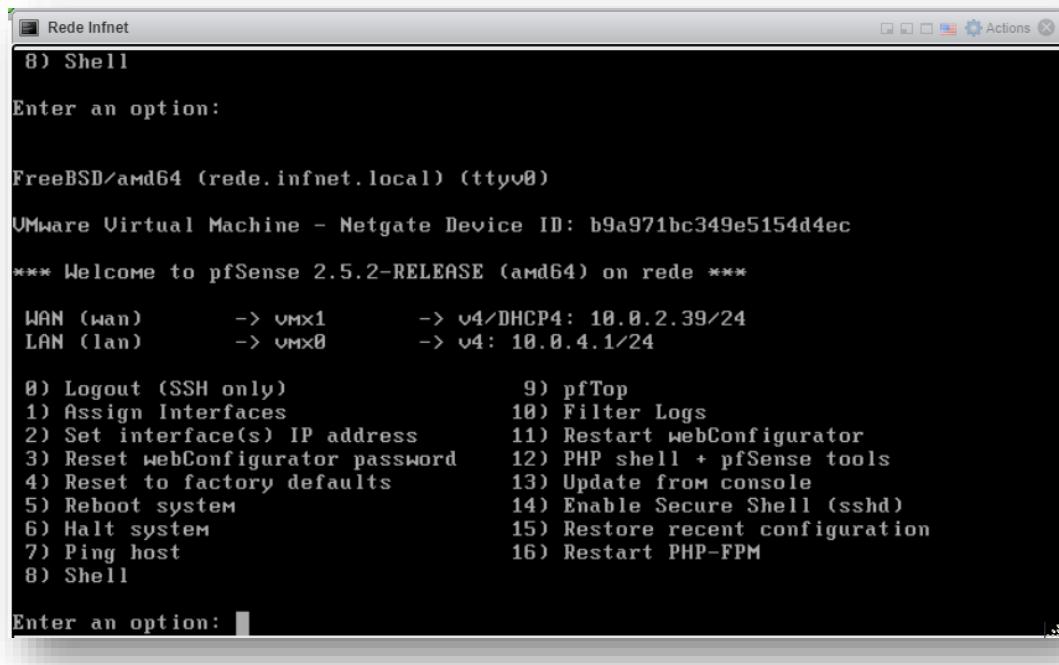


Figura 44 – Máquina Virtual PFsense

Após realizar a instalação padrão a console da máquina virtual apresenta o menu conforme a próxima imagem:



```

Rede Infnet
8) Shell

Enter an option:

FreeBSD/amd64 (rede.infnet.local) (ttyv0)

VMware Virtual Machine - Netgate Device ID: b9a971bc349e5154d4ec

*** Welcome to pfSense 2.5.2-RELEASE (amd64) on rede ***
WAN (wan)      -> vmx1      -> v4/DHCP4: 10.0.2.39/24
LAN (lan)      -> vmx0      -> v4: 10.0.4.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults   13) Update from console
5) Reboot system               14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: [■]

```

Figura 45 – Tela inicial console PFSense

Neste menu é possível definir VLans (Opção 1), função das interfaces de rede (Opção 1), configuração do tipo e qual endereçamento de cada interface (Opção 2), entre outros recursos técnicos relacionados a operação e manutenção da aplicação.

Na imagem a seguir é demonstrado o Dashboard do PFSense, demonstrando as seguintes informações: Informações do Sistema, consumo de recursos, versionamento, configuração DNS, configuração das Interfaces, monitoramento do Gateway, monitoramento de tráfego das interfaces, além dos menus de configuração. Segue imagem:

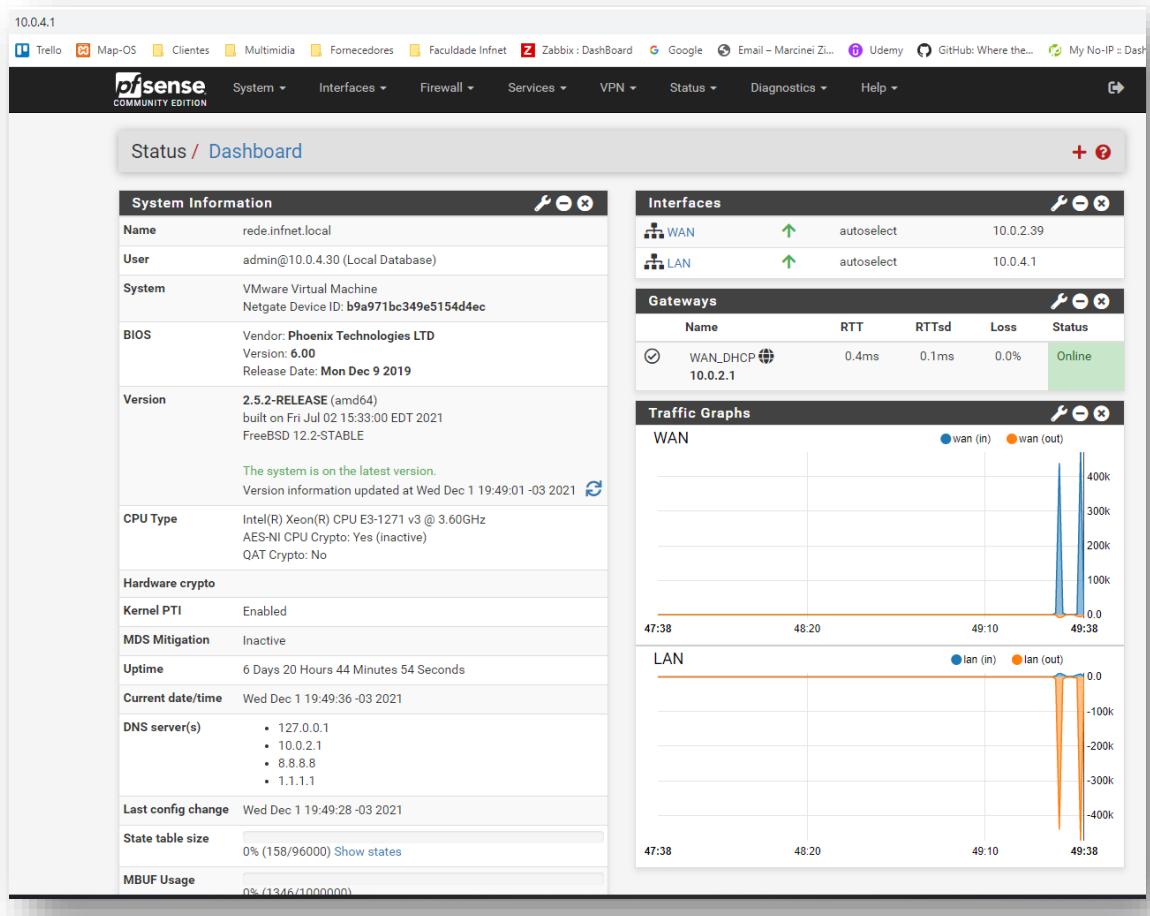


Figura 46 – Dashboard PFSense

Além da função de roteador, o principal recurso da aplicação é fornecer endereçamento IP via DHCP e criar tabela contendo os endereços estáticos. Esta solução é fundamental para evitar conflitos de IP e por consequência cria identificação e organização dos equipamentos pertencentes a estrutura. Segue imagem com a tabela de identificação dos hosts.

The screenshot shows the pfSense web interface at the URL 10.0.4.1/status_dhcp_leases.php. The top navigation bar includes links to various services like Trello, Map-OS, Clientes, Multimedia, Fornecedores, Faculdade Infnet, Zabbix Dashboard, Google, Email, Udemy, GitHub, and My No-IP Dashboard. The main title is "Status / DHCP Leases". Below the title is a search bar with a placeholder "Search term" and a dropdown set to "All". There are "Search" and "Clear" buttons. A note below the search bar says "Enter a search string or *nix regular expression to filter entries." The main content is a table titled "Leases" with the following columns: IP address, MAC address, Client Id, Hostname, Description, Start, End, Online, Lease Type, and Actions. The table contains four rows of data:

IP address	MAC address	Client Id	Hostname	Description	Start	End	Online	Lease Type	Actions
10.0.4.6	00:0c:29:5d:17:fa	app	app	VM App PB	n/a	n/a	online	static	
10.0.4.5	00:0c:29:20:c0:cf	freenas	freenas	Storage Freenas Infnet	n/a	n/a	online	static	
10.0.4.4	00:0c:29:8c:c9:28	ansible	ansible	VM Ansible PB	n/a	n/a	online	static	
10.0.4.30	48:ba:4e:bb:21:4e		Marcinei-HP		2021/12/01 22:41:33	2021/12/02 00:41:33	online	active	

Figura 47 – Tabela de reserva de endereços do serviço DHCP

Após esta etapa é assegurado o endereçamento dos hosts e acesso a rede e internet por parte deles. O acesso a internet é fundamental para aplicar atualizações, download de recursos, comunicação entre os hosts, e principalmente permite acesso ao portal Git para execução da automação de implantação do Ansible.

7.7.3 Armazenamento

Será demonstrado as etapas para a criação do datastore a ser utilizado pela máquina virtual App. A intenção do uso deste recurso é buscar mostrar de maneira simplificada a aplicação da metodologia de armazenamento utilizada em estruturas de datacenter. Nesta visão o(s) datastore(s) são armazenados em equipamento com recursos corporativos e que fornecem grande redundância e segurança no armazenamento. Com a utilização de compartilhamentos NFS (Network File System) é possível adicionar ao virtualizador datastores compartilhados.

A criação da máquina virtual definiu as seguintes configurações:

- Processador: 1 vCPU
- Memória: 4GB

- HD: 5 unidades de 20GB cada
- Interface de rede: 1
- Sistema Operacional: FreeNAS 11.3 U5

Segue tela com as definições da máquina virtual:

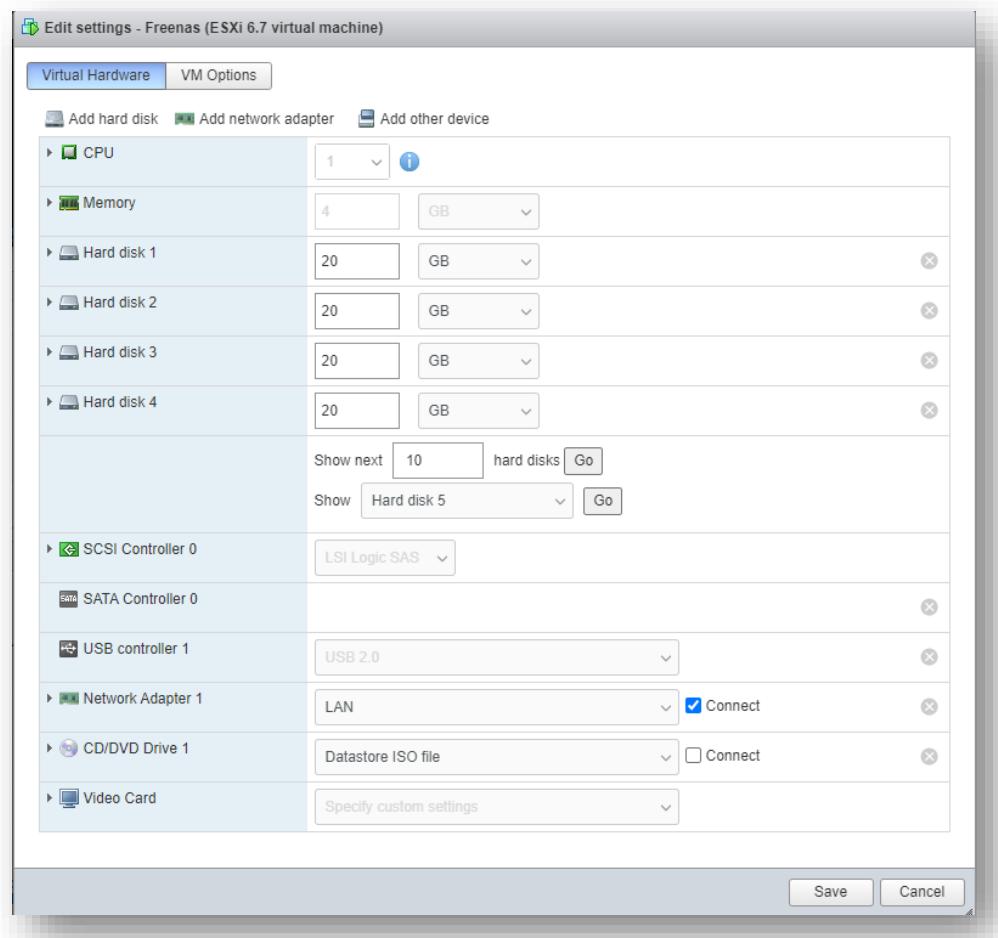


Figura 48 – Detalhes da configuração da Máquina Virtual FreeNAS

A instalação teve início a partir do download da imagem do instalador oficial fornecido pelo próprio desenvolvedor, sendo este realizado do seguinte site:

<https://download.freenas.org/11.3/STABLE/U5/x64/FreeNAS-11.3-U5.iso>

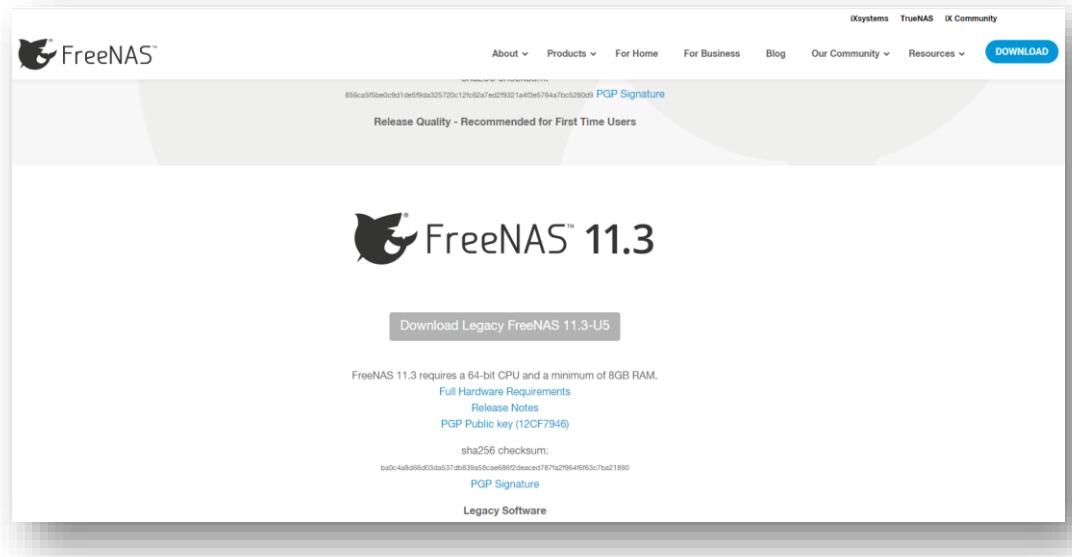


Figura 49 – Site para download da imagem do instalador oficial FreeNAS 11.3-U5

Após realizar o download é iniciado a instalação seguindo a tela que permite realizar a instalação ou upgrade, utilizar o shell de comandos, reiniciar o sistema ou desligar a máquina. Segue imagem:

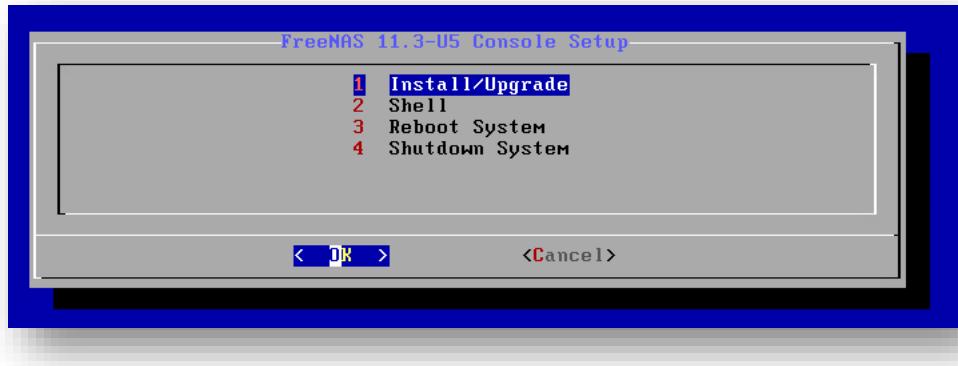


Figura 50 – Início da instalação do FreeNAS

Na segunda tela é demonstrado os discos rígidos disponíveis para instalação da aplicação FreeNAS, sendo que será selecionado com a tela de “espaço” o primeiro disco. Este disco não fará parte do armazenamento disponível a rede ou ao usuário, será utilizado somente pela aplicação de Storage. Segue imagem:

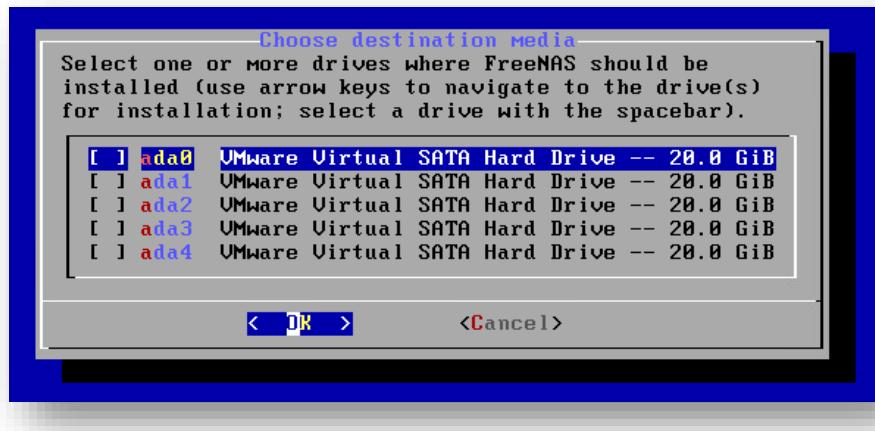


Figura 51 – Seleção do disco onde será instalado o FreeNAS

Será exibido uma tela de advertência informando que os dados armazenados no disco serão perdidos, segue imagem:



Figura 52 – Advertência sobre perda de dados no disco

Na próxima etapa será solicitado a senha para acesso do usuário root a aplicação, segue captura da tela:

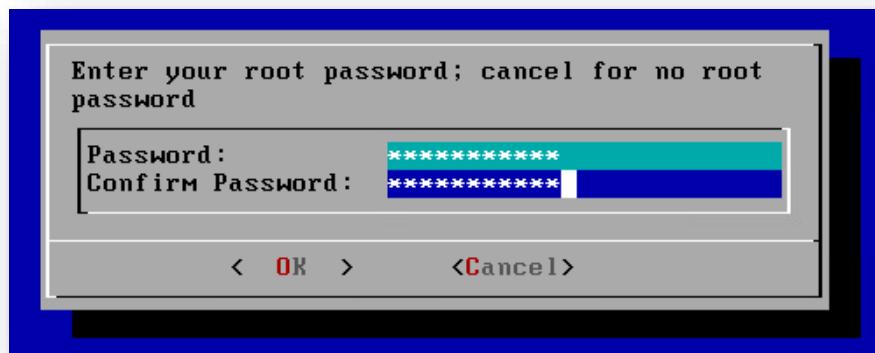


Figura 53 – Criação de senha para acesso

O instalador questionará e informará que não é possível realizar o boot quando o modo selecionado é o UEFI, sendo realizado somente no modo BIOS. Segue:

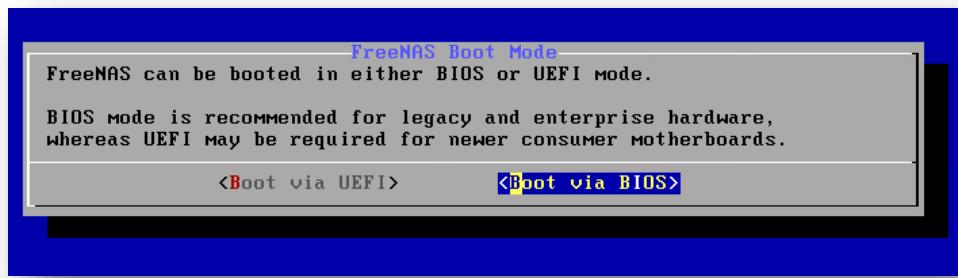


Figura 54 – Confirmação do modo de Boot

Será informado a finalização com sucesso da instalação do FreeNAS e solicitado que seja retirado a mídia de instalação da aplicação. No VMWare ESXi é necessário alterar no dispositivo CD/DVD para a opção Host Device, desta forma evitando a possibilidade de realizar a inicialização da máquina virtual pela imagem do instalador novamente. Segue imagem:



Figura 55 – Tela de finalização da instalação

Ao iniciar o FreeNAS é demonstrado opções para configuração da rede, senhas de acesso, acesso ao shell de comando e opções de desligamento ou reboot. Para utilizar os recursos completos do appliance é necessário realizar o login em um dos endereços determinados abaixo da linha “The web user interface is at:”, que neste caso é:

<http://10.0.4.5>

<https://10.0.4.5>

Lembrando que este endereço do FreeNAS foi definido pelo servidor DHCP da aplicação PFSense, através de reserva de endereço IP pelo endereço MAC da interface de rede desta máquina virtual.



```
Freenas
https://10.0.4.5
Enter an option from 1-11:
Console setup
-----
1) Configure Network Interfaces
2) Configure Link Aggregation
3) Configure VLAN Interface
4) Configure Default Route
5) Configure Static Routes
6) Configure DNS
7) Reset Root Password
8) Reset Configuration to Defaults
9) Shell
10) Reboot
11) Shut Down

The web user interface is at:
http://10.0.4.5
https://10.0.4.5
Enter an option from 1-11: █
```

Figura 56 – Tela da console da Máquina Virtual do FreeNAS

Para realizar o acesso é necessário executar um navegador Web e inserir o endereço do servidor Storage FreeNAS. Segue abaixo tela de login:

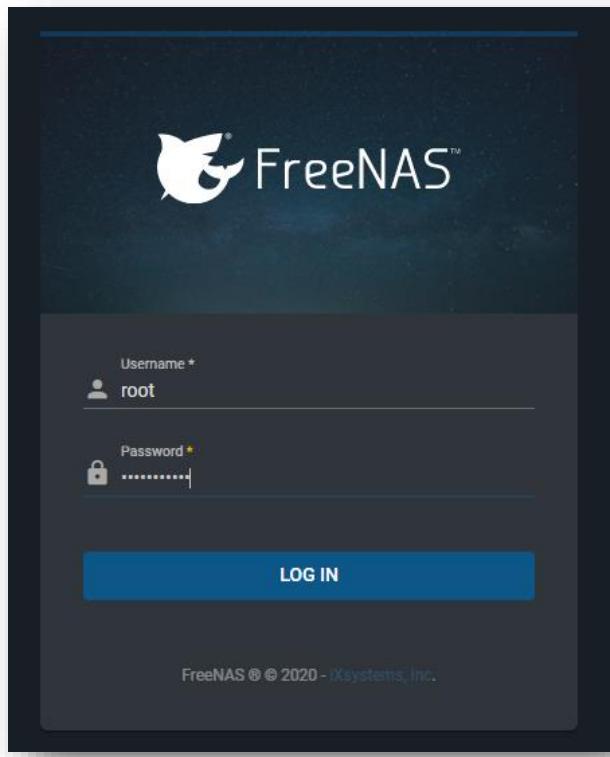


Figura 57 – Tela de login

Abaixo está demonstrada a tela principal do appliance onde constam informações do sistema FreeNAS, utilização de recurso de processador, Memória e do Pool de armazenamento criado. Segue:

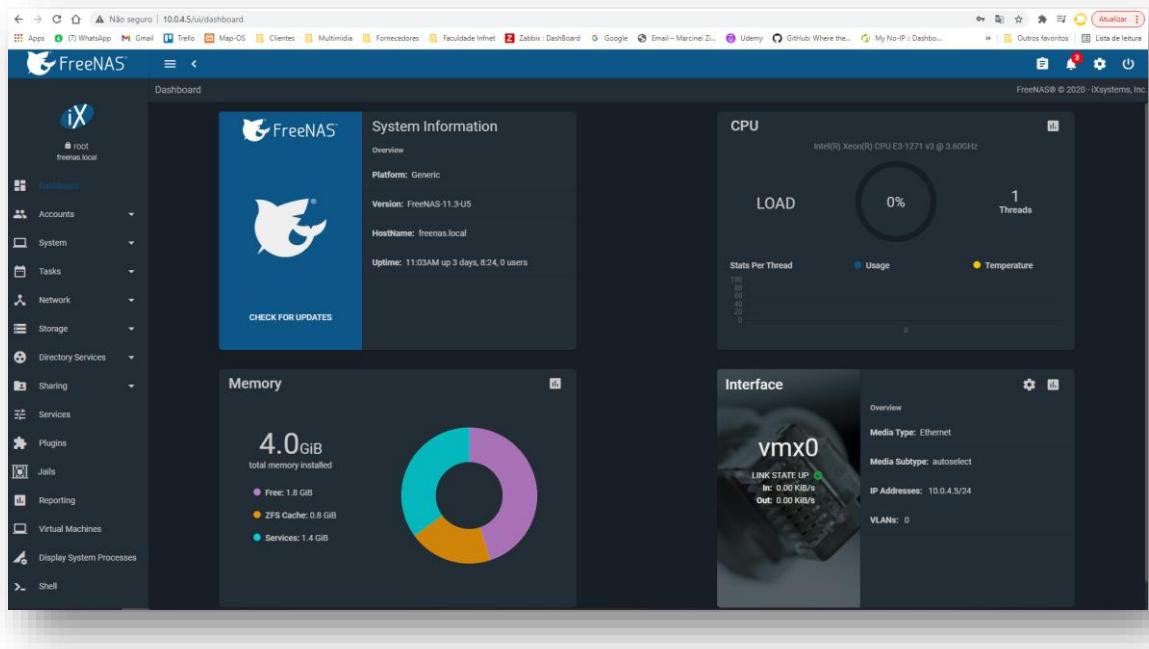


Figura 58 – Dashboard FreeNas

A primeira etapa é a criação dos Pool's de armazenamento, que consiste em criar um array composto pelos 4 discos adicionados a máquina virtual, sendo eles de capacidade de 20GB. Para a criação é preciso acessar o menu “Storage” na barra lateral esquerda, em seguida a opção “Pools”. Selecionar a opção “ADD” no canto direito superior e na próxima tela selecionar a opção “Create new pool” e clicar em “Create Pool”. Segue imagem:

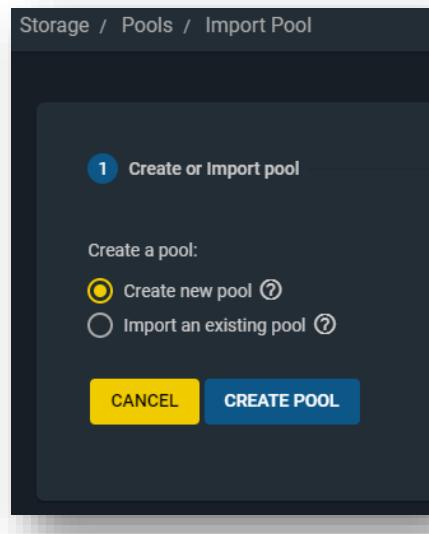


Figura 59 – Criação de Pool

Insira o nome no campo “Name”, neste caso será “datastoreA”. Na parte “Available Disks” selecionar os discos que farão parte do array (pool), que neste caso serão da1, da2, da3 e da4 para este datastorer. Clicar em seguida na seta azul apontando para a parte “Data VDevs”. Na opção abaixo de “Data VDevs” é seleciona a opção “Raid-z2”, que consiste na configuração semelhante ao raid RAID 6. Como os discos de 20GB equivalem em torno de 18,6GB em armazenamento real o total de armazenamento seria de 74,4GB, mas como está utilizado o Raid-z2 (semelhante ao Raid 2), a capacidade final é de 36 GB aproximadamente.

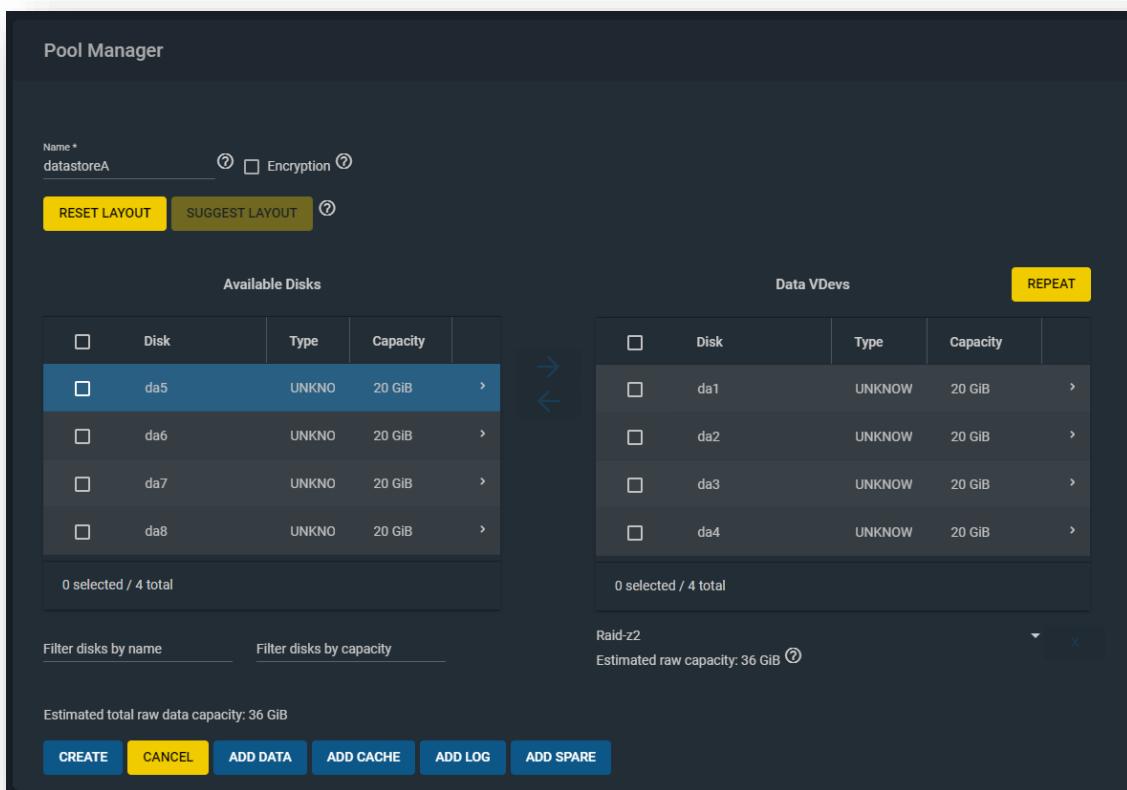


Figura 60 – Criação do Array DatastoreA

Na imagem abaixo é demonstrado o array “datastoreA” criado e de maneira operacional, contendo informações de armazenamento e espaço livre, segue imagem:

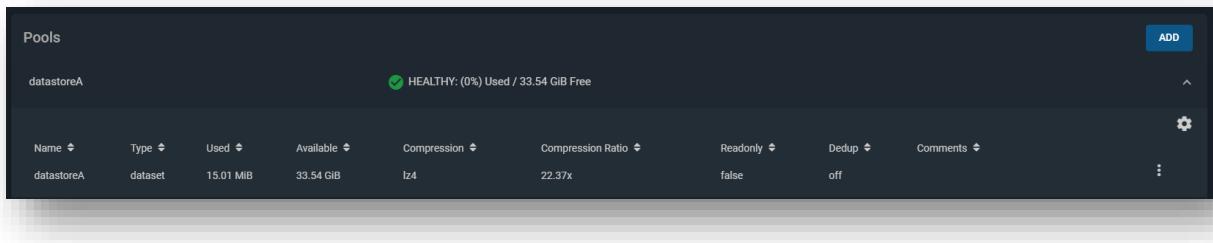


Figura 61 – Datastore “datastoreA”

A próxima etapa é criar o compartilhamento do volume NFS a ser adicionado posteriormente pelo VMWare ESXi. Abaixo está identificado no meu a opção “Sharing” e o tipo “Unix Shares (NFS)”, segue imagem:

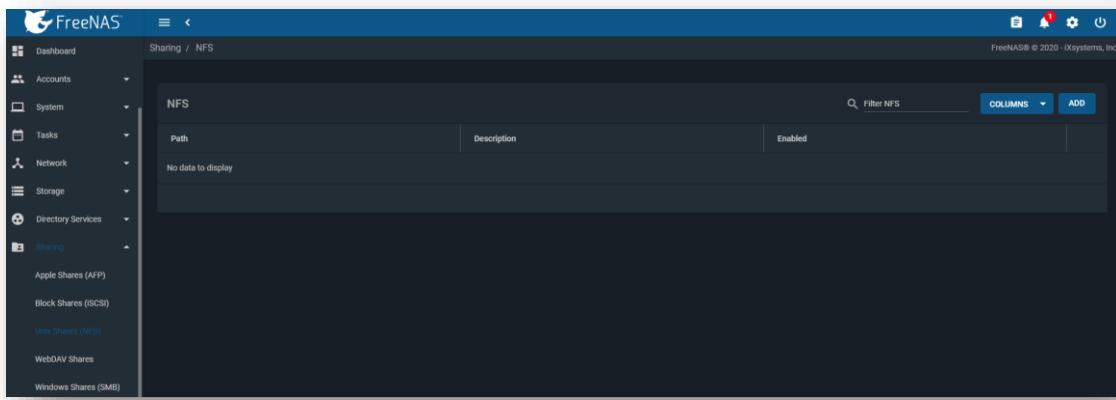


Figura 62 – Menus a serem selecionados para criação do compartilhamento NFS

Ao selecionar no canto direito superior a opção “ADD” será exibida a página abaixo com as configurações para criação. Nesta etapa será necessário definir inicialmente o caminho onde está montado o armazenamento (`/mnt/datastoreA`) e em seguida habilitar as opções “All dirs” (permite que as pastas e sub pastas façam parte do compartilhamento) e “Enabled”. Também é preciso informar a rede autorizada (10.0.4.0/24) e o endereço IP do ESXi que irá utilizar este compartilhamento (10.0.4.3). Por último é necessário definir o usuário que irá administrar o compartilhamento (root) e o grupo (neste caso não há grupo definido: nogroup). Seguem imagens demonstrando a criação do compartilhamento do datastore:

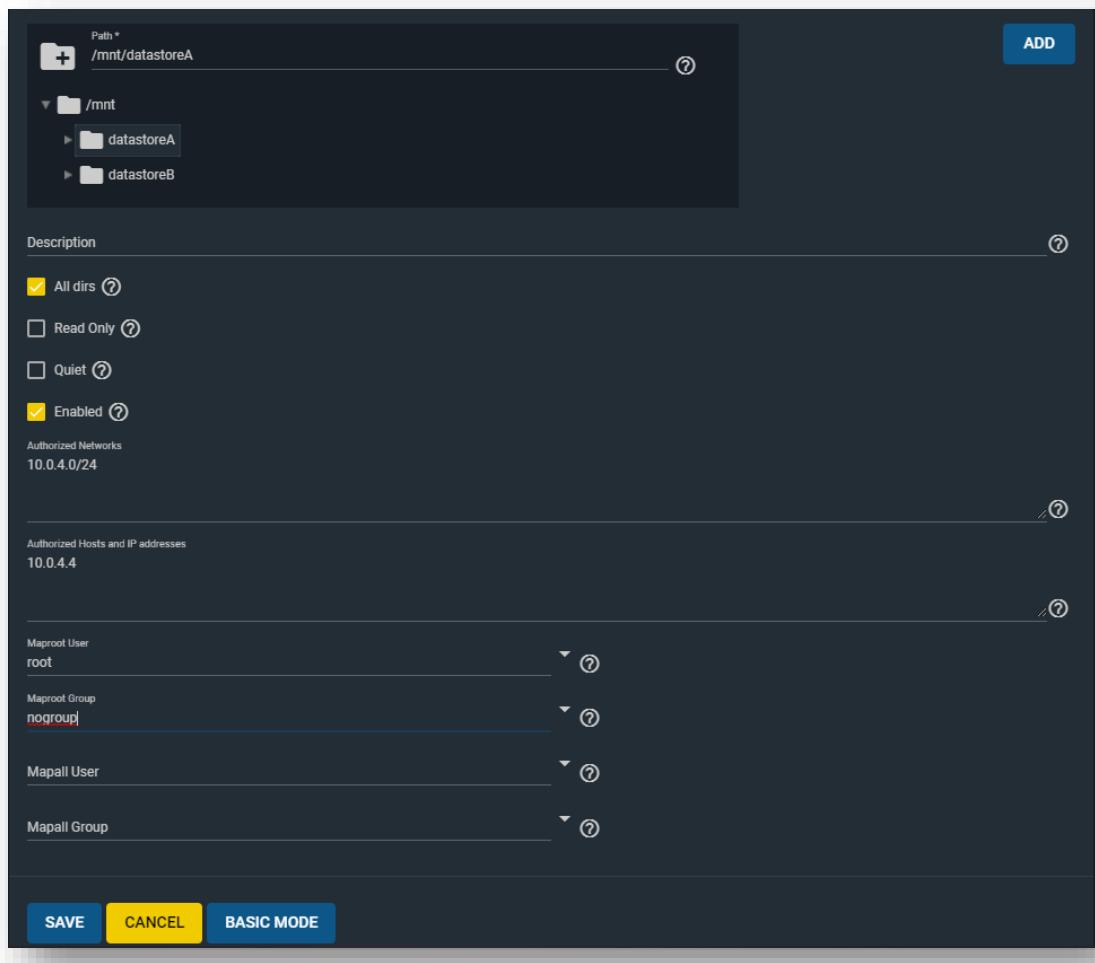


Figura 63 – Criação do compartilhamento NFS

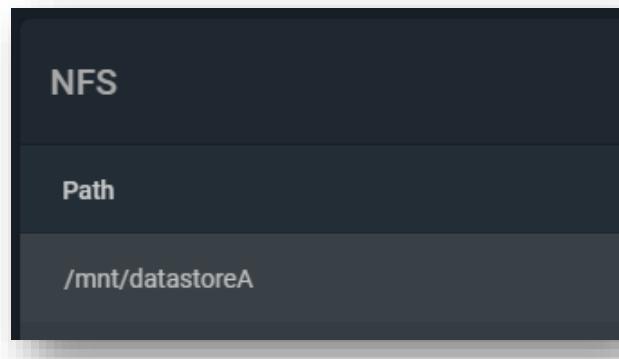


Figura 64 – Compartilhamento NFS habilitado

A adição do datastore FreeNAS ao ESXi é demonstrado nas próximas capturas de tela. A primeira etapa é selecionar o tipo “Mount NFS datastore”, segue imagem:

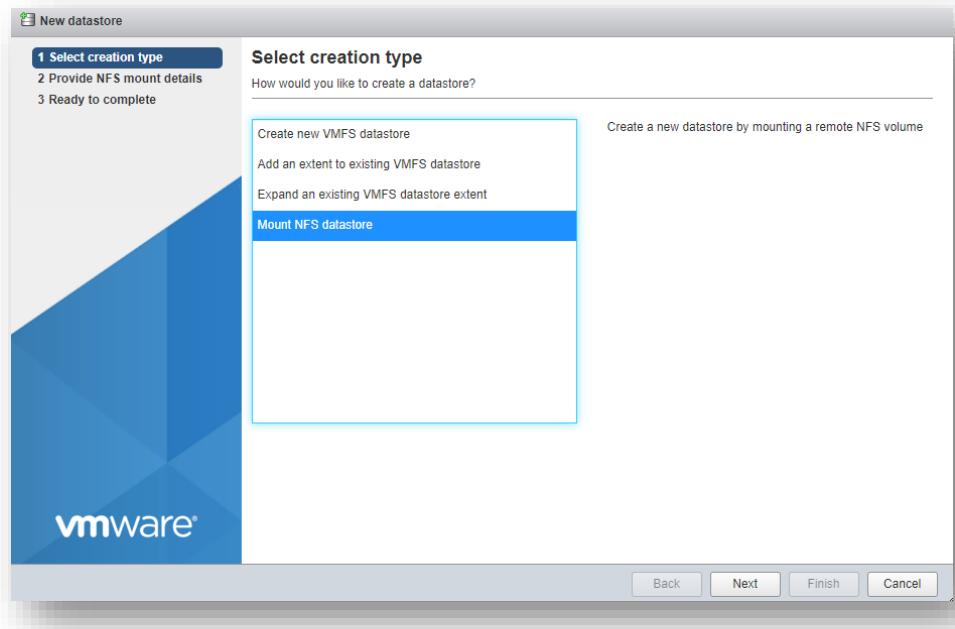


Figura 65 – Seleção do tipo de Datastore “Mount NFS Datastore”

As configurações específicas são informadas na segunda etapa, onde é preciso determinar o nome do Datastore (datastoreA), o endereço IP do servidor Storage FreeNAS (10.0.4.5), o caminho do compartilhamento (/mnt/datastoreA) e por último a versão. Seguem capturas de tela:

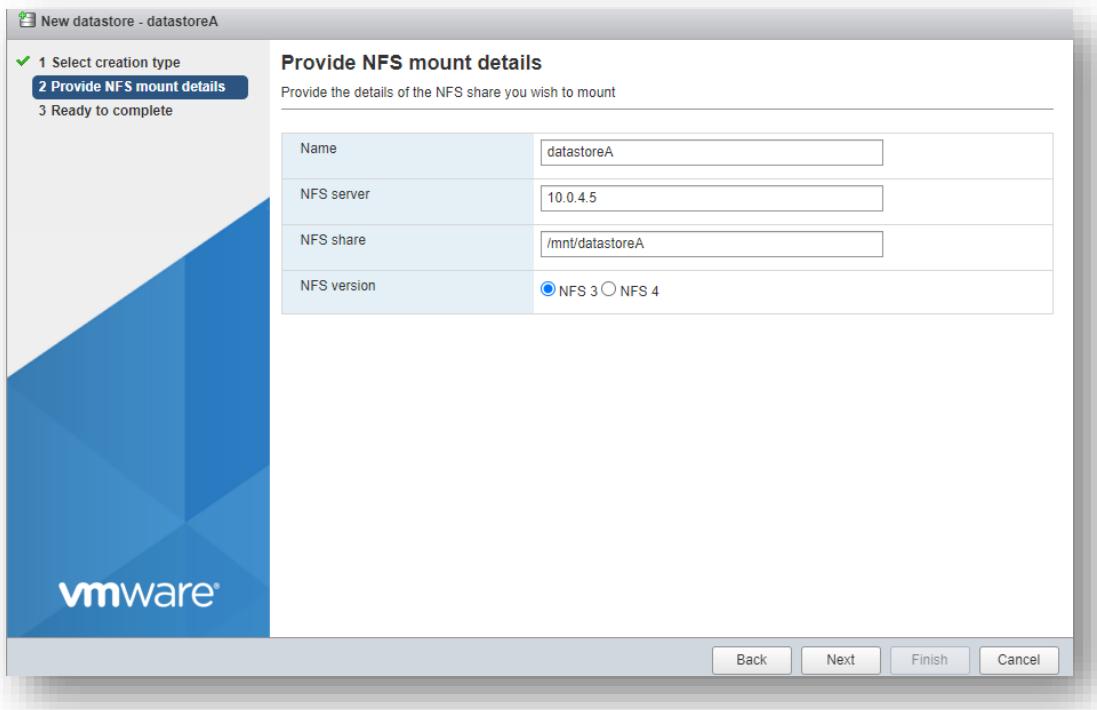


Figura 66 – Inserindo informações do compartilhamento NFS

Na imagem a seguir é apresentado informações do datastore adicionado, segue:

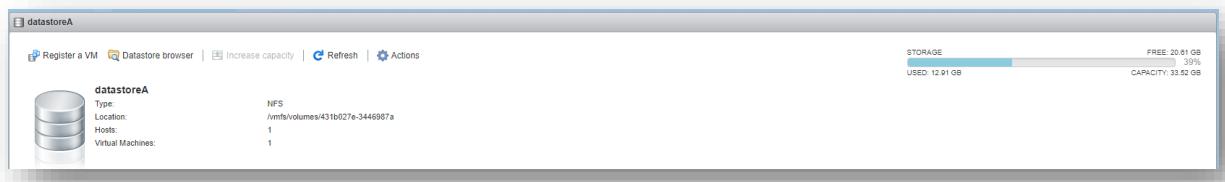


Figura 67 – Informações do datastore “datastoreA”

Na tela a seguir é demonstrado a utilização da máquina virtual “UbuntuAPP”, onde será executado a aplicação. Segue imagem:

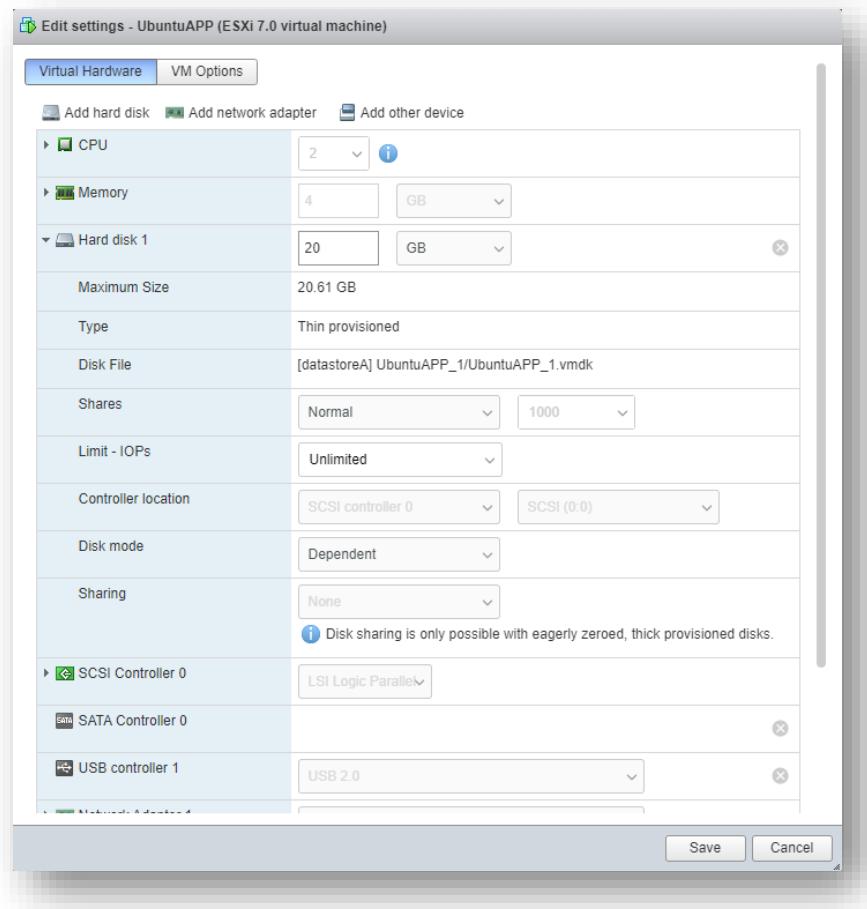


Figura 68 – Utilização do datastore NFS

Finalizada esta etapa é possível realizar a instalação dos sistemas operacionais e demais configuração e implantações.

7.7.4 Sistemas Operacionais

Após preparar a estrutura é possível realizar a instalação do sistema operacional das máquinas virtuais “Ubuntu-Ansible” (execução do ansible) e “UbuntuAPP” (execução da aplicação WordPress).

Para a realização da implantação é utilizado duas máquinas virtuais utilizando o sistema operacional Linux com a distribuição Ubuntu na versão 20.04, com os seguintes recursos computacionais em cada uma:

- 02 vCPUs
- 04 GB de memória
- 50GB de armazenamento (Ansible) e 20GB (App)
- 01 interface de rede

Abaixo está capturada tela com as etapas de criação da máquina virtual “UbuntuApp”, a qual será semelhante a VM “Ubuntu-Ansible”, alterando somente o armazenamento. Seguem imagens:

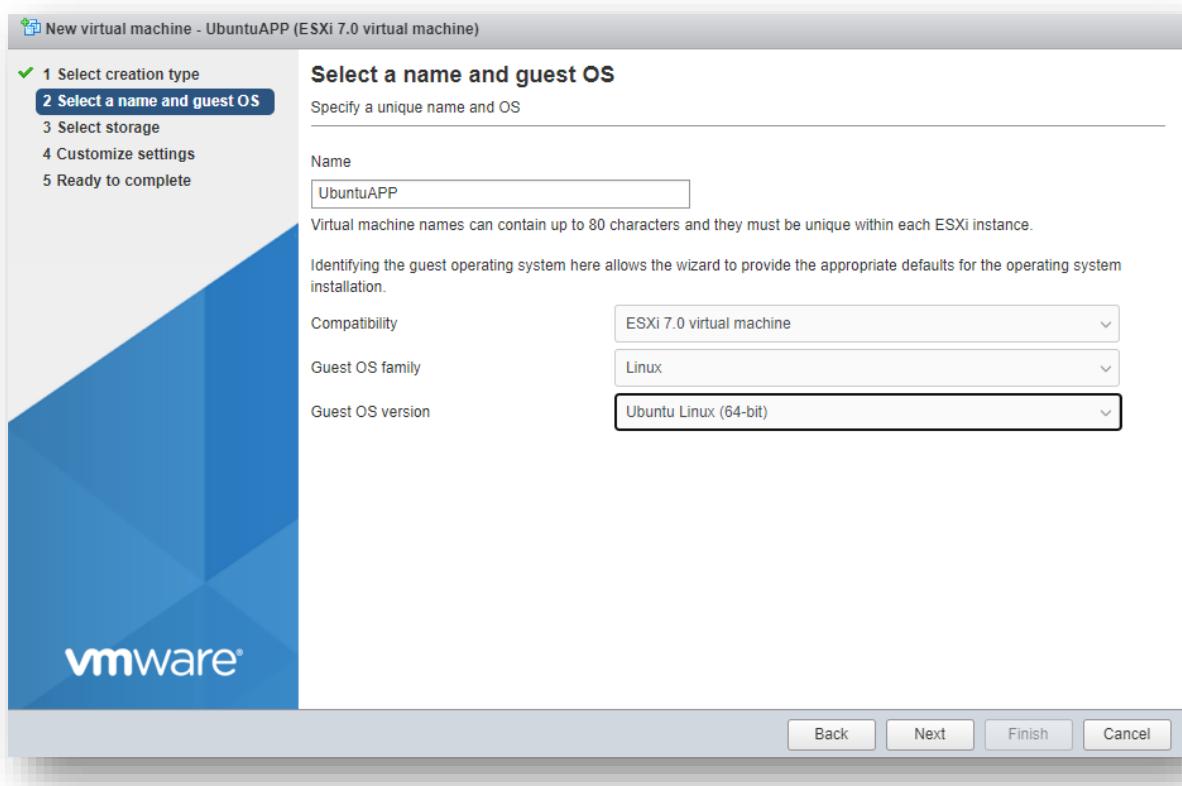


Figura 69 – Criação da VM, inserindo o nome, compatibilidade e sistema operacional

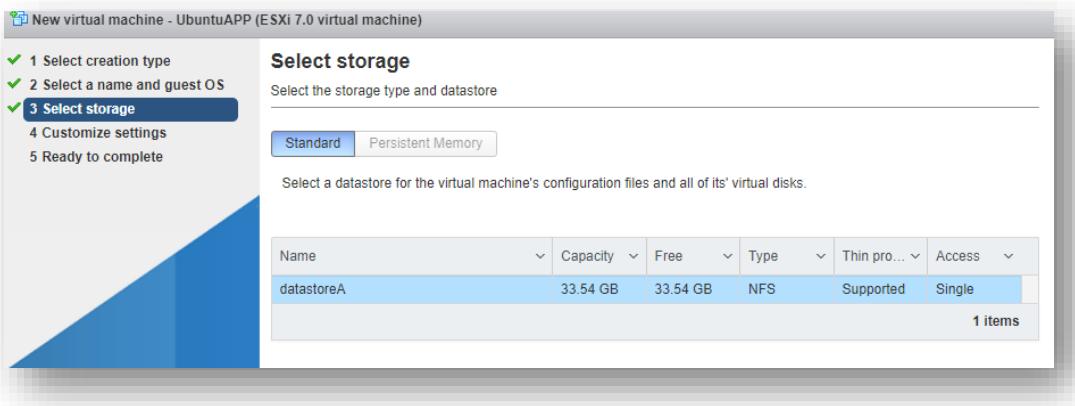


Figura 70 – Escolha do datastore

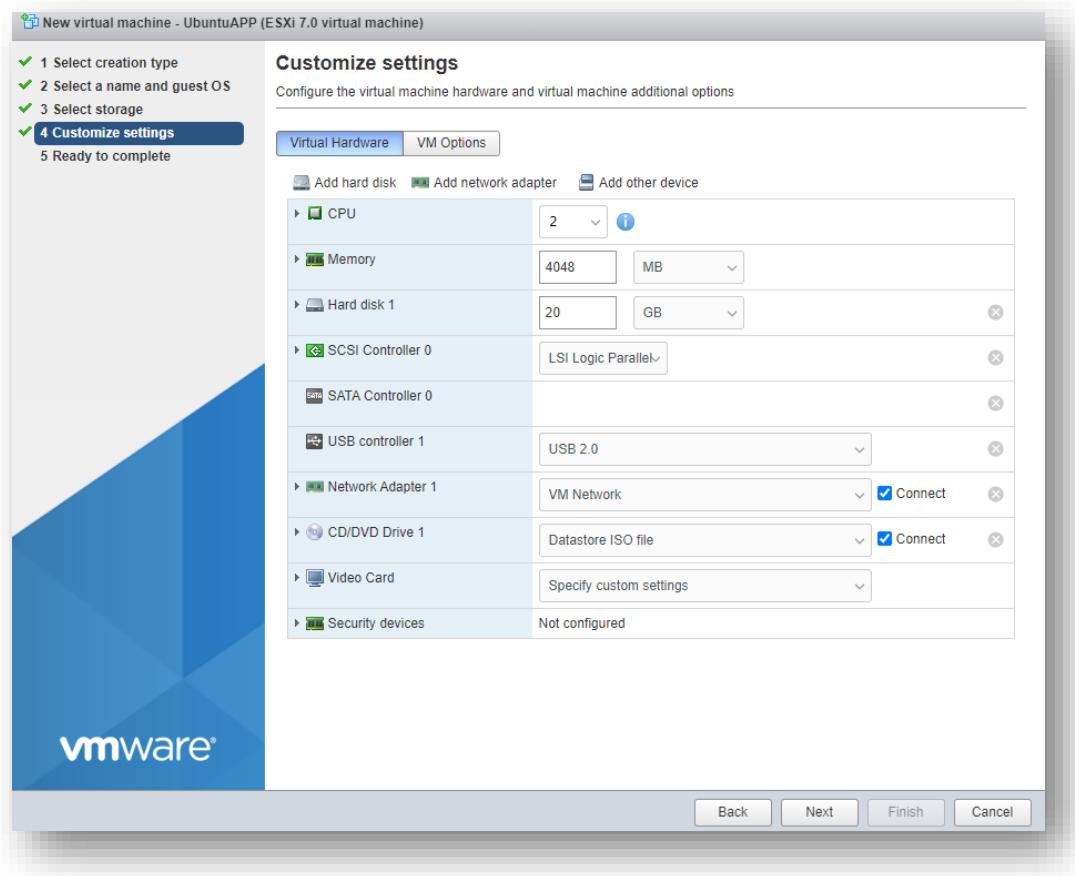


Figura 71 – Definição das demais especificações da máquina virtual

Após finalizar a criação da VM é possível ligar a mesma. Na tela abaixo é demonstrado as etapas da instalação do sistema operacional Ubuntu 20.04, segue:



Figura 72 – Seleção do tipo de instalação (Live ou Instalação completa) e idioma

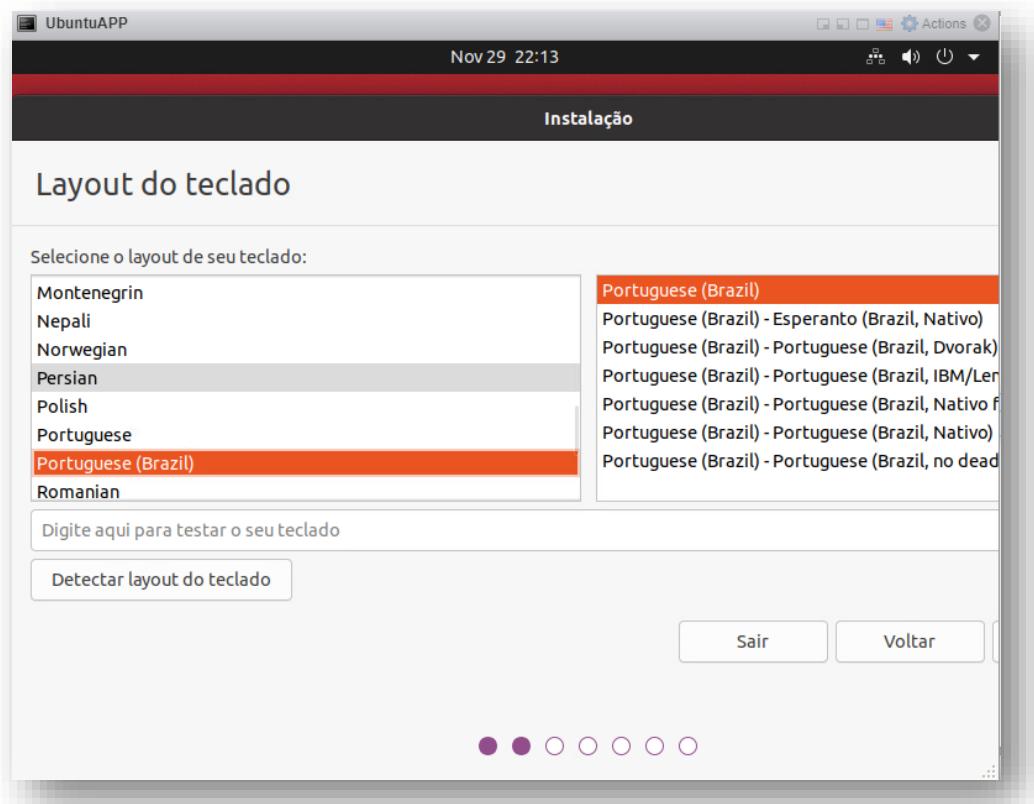


Figura 73 – Escolha do layout e idioma do teclado

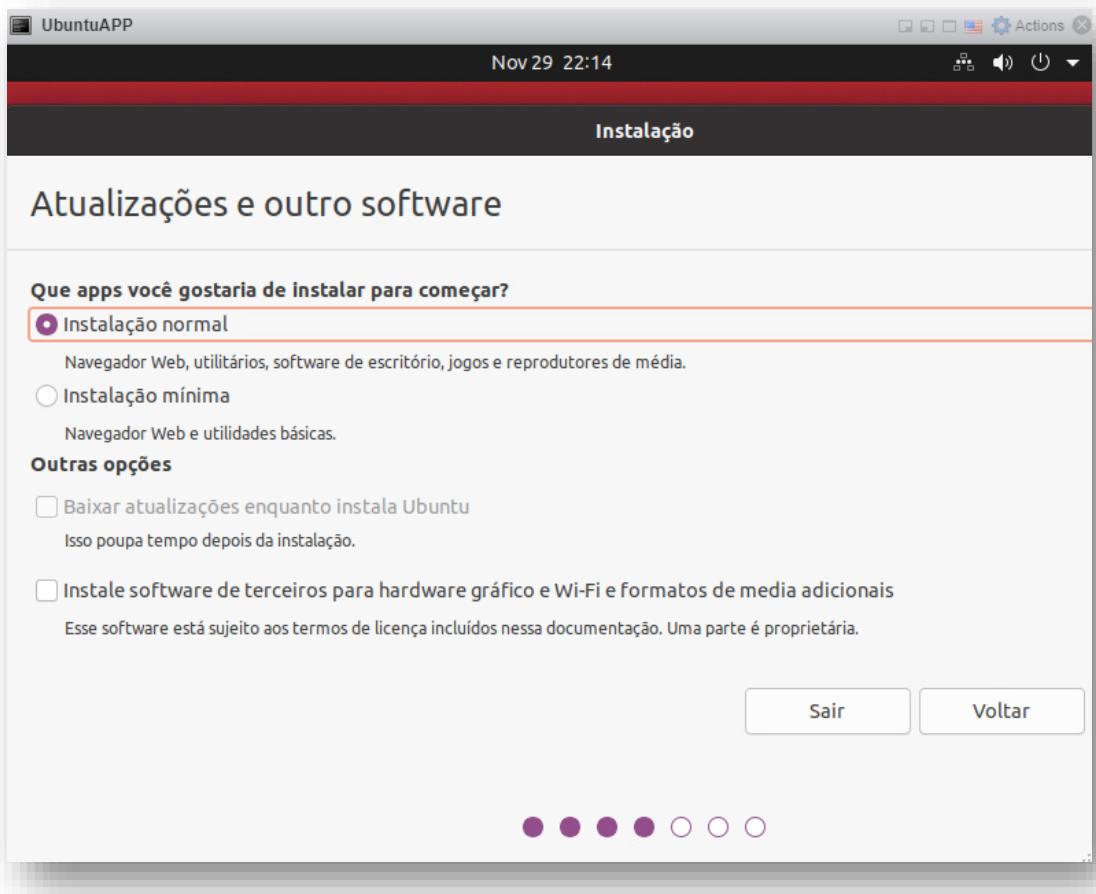


Figura 74 – Tipo de instalação e recursos

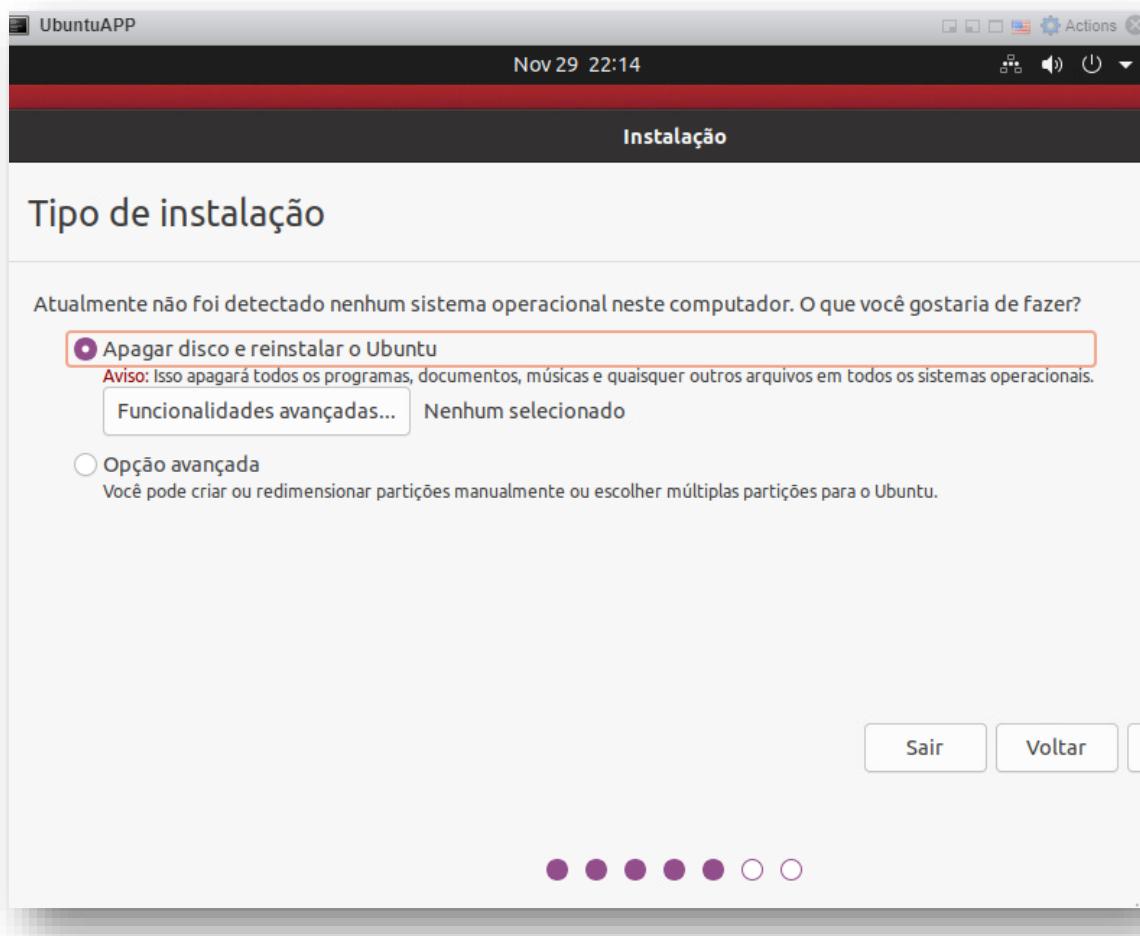


Figura 75 – Utilitário de gerenciamento do disco e opções de instalação

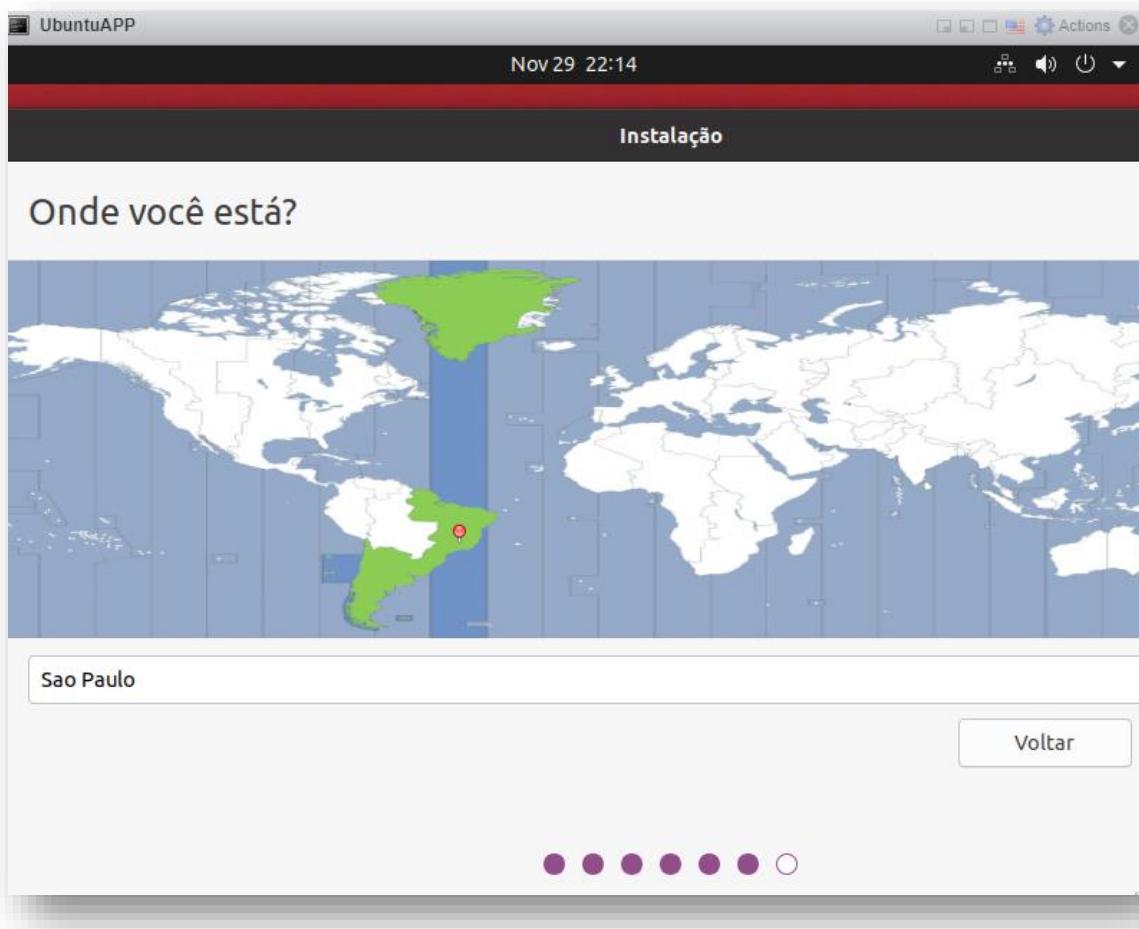


Figura 76 – Seleção do fuso horário

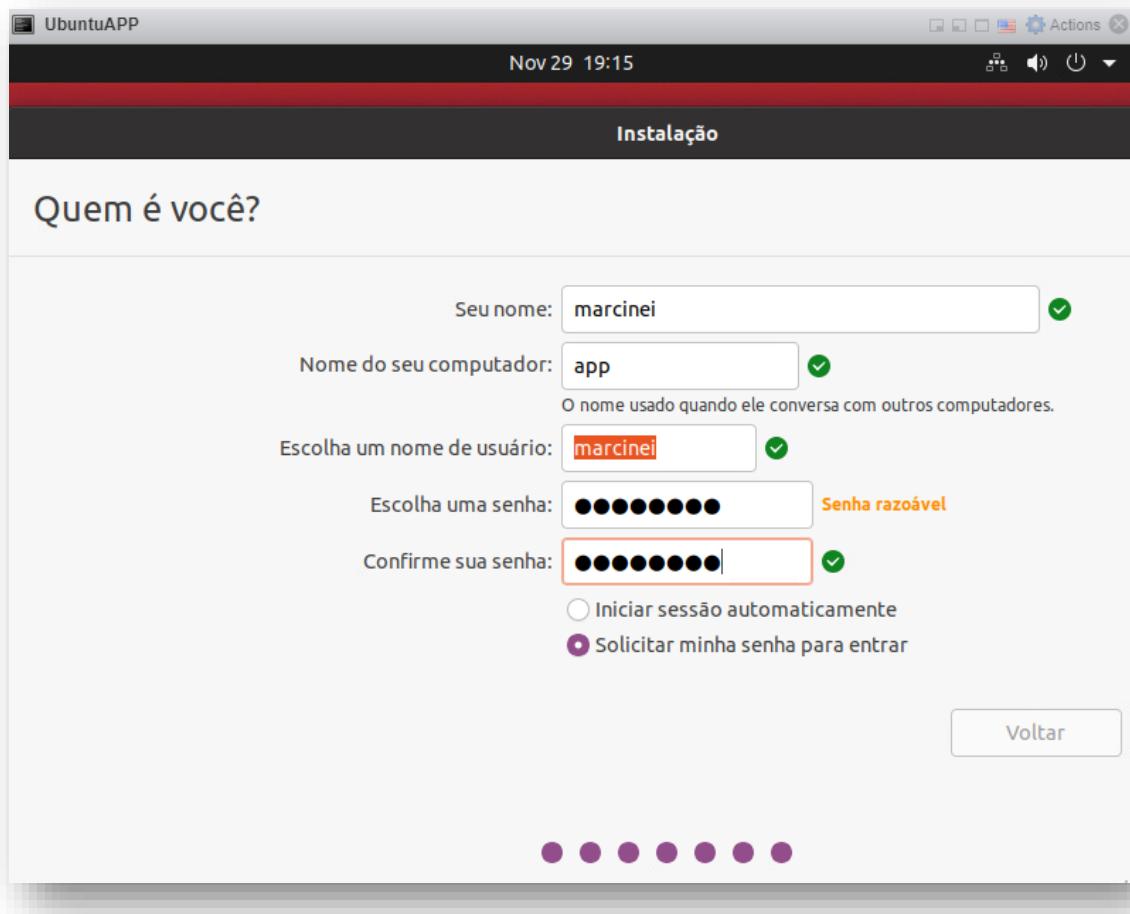


Figura 77 – Configuração do host, usuário e senha de acesso

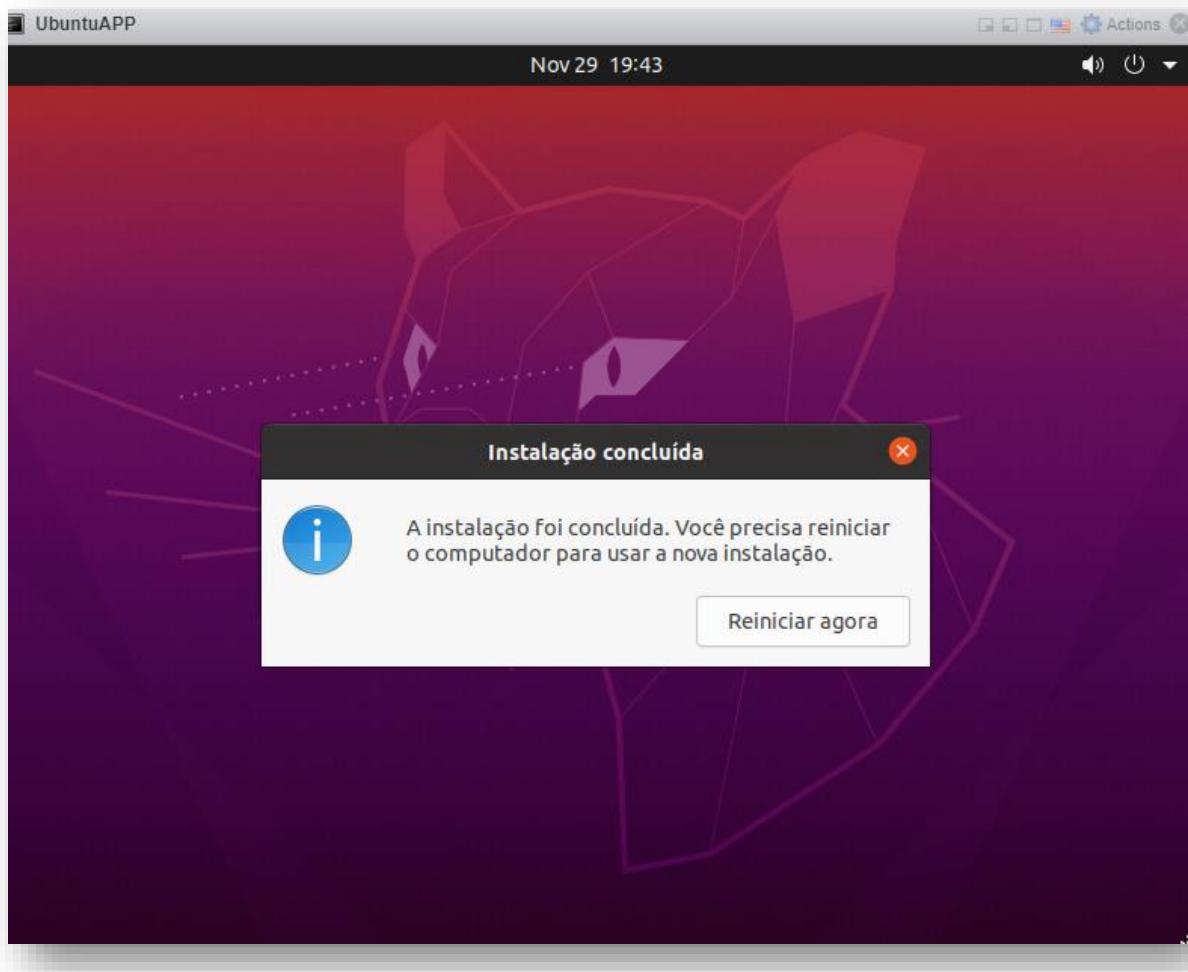


Figura 78 – Conclusão da instalação e indicação de reinício do sistema operacional

Após a conclusão da instalação e primeira inicialização o próprio sistema operacional indica a atualizações. Esta operação é indicada e pode ser feita também através do terminal utilizando o seguinte comando:

```
sudo apt update
```

Segue imagem com a tela inicial do sistema operacional em execução.

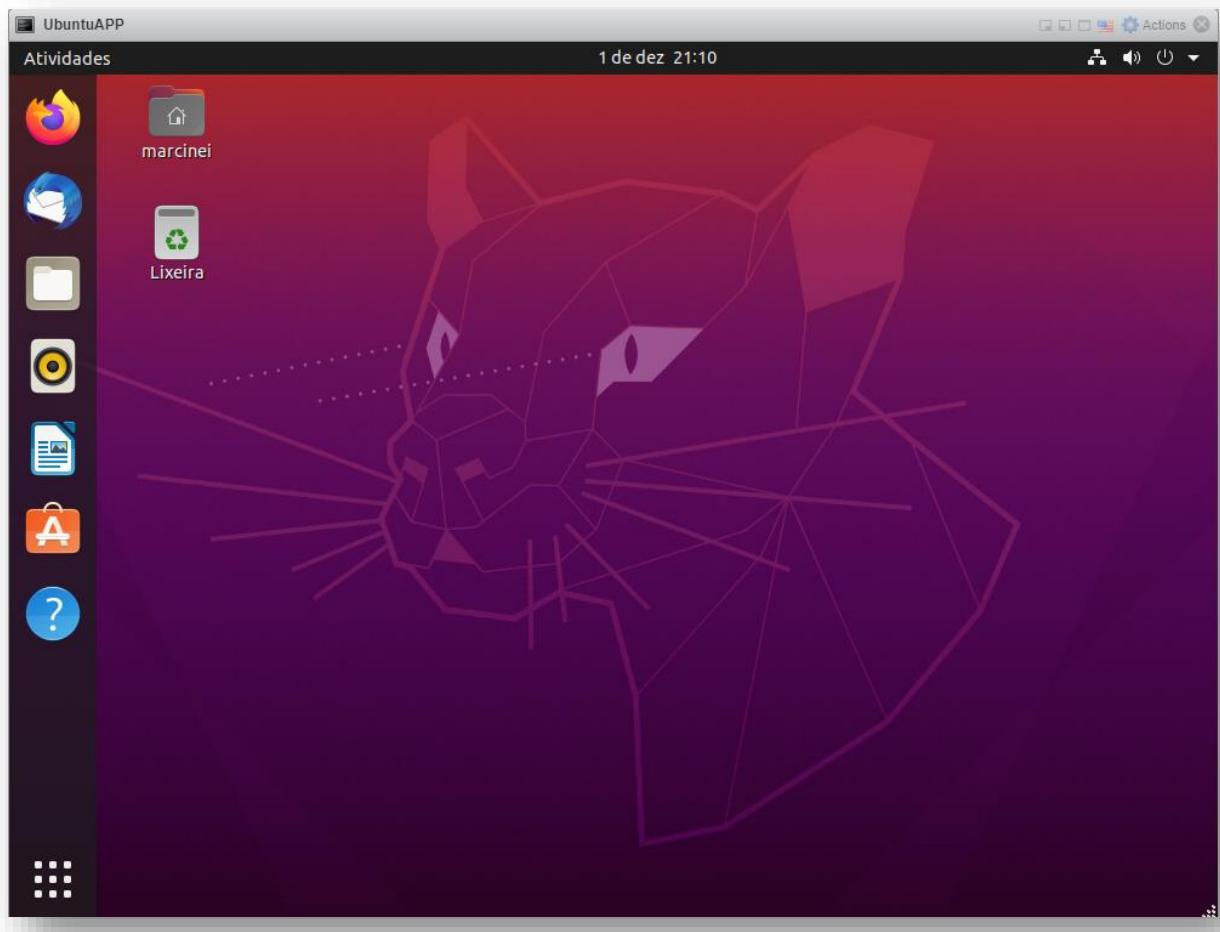


Figura 79 – Tela inicial do sistema operacional Linux Ubuntu 20.04

Finalizada a instalação das duas máquinas virtuais é possível iniciar a implantação dos recursos necessários para o WordPress.

7.7.5 Ansible

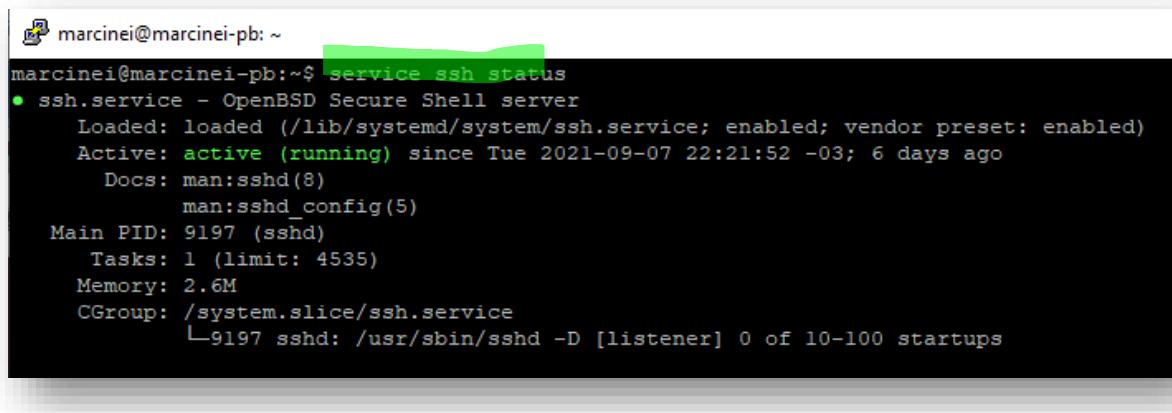
A primeira etapa após a instalação padrão do sistema operacional Ubuntu é instalar o servidor SSH, a qual permitirá acesso remoto para configurações locais e posteriormente integração com a ferramenta de automação Ansible. As etapas relacionadas ao acesso SSH

serão realizadas nas duas máquinas virtuais Ubuntu. Segue comando para instalação e em seguida para verificação do seu estado:

```
sudo apt install openssh-server
```

```
service ssh status
```

Segue imagem com a demonstração do estado de funcionamento do serviço de SSH:



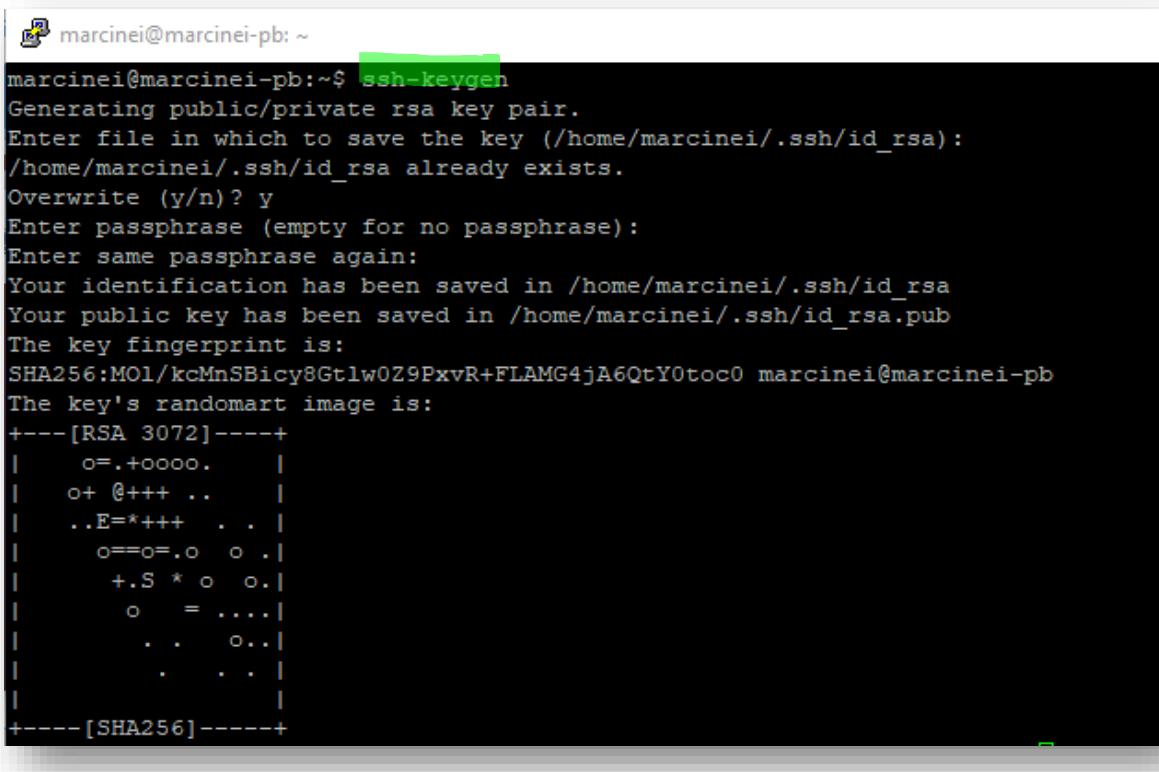
```
marcinei@marcinei-pb: ~$ service ssh status
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2021-09-07 22:21:52 -03; 6 days ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 9197 (sshd)
     Tasks: 1 (limit: 4535)
   Memory: 2.6M
      CGroup: /system.slice/ssh.service
              └─9197 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

Figura 80 – Serviço SSH em funcionamento

Para que seja estabelecido logon automaticamente no acesso SSH a estações Linux é realizado a criação de uma chave SSH, a qual permitirá o login por meio desta, sem a necessidade de informar usuário e senha. Segue comando utilizado:

```
ssh-keygen
```

Segue imagem capturada com informações do processo de criação desta chave:



```

marcinei@marcinei-pb: ~
marcinei@marcinei-pb:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/marcinei/.ssh/id_rsa):
/home/marcinei/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/marcinei/.ssh/id_rsa
Your public key has been saved in /home/marcinei/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:MO1/kcMnSBicy8Gtlw0Z9PqvR+FLAMG4jA6QtY0toc0 marcinei@marcinei-pb
The key's randomart image is:
+---[RSA 3072]---+
|   o=+oooo. |
|   o+ @+++ .. |
|   ..E=**+++. . |
|   o==o=.o o .|
|   +.S * o o.|
|   o   = ....|
|   . .   o...|
|   . .   . . |
|                         |
+---[SHA256]---+

```

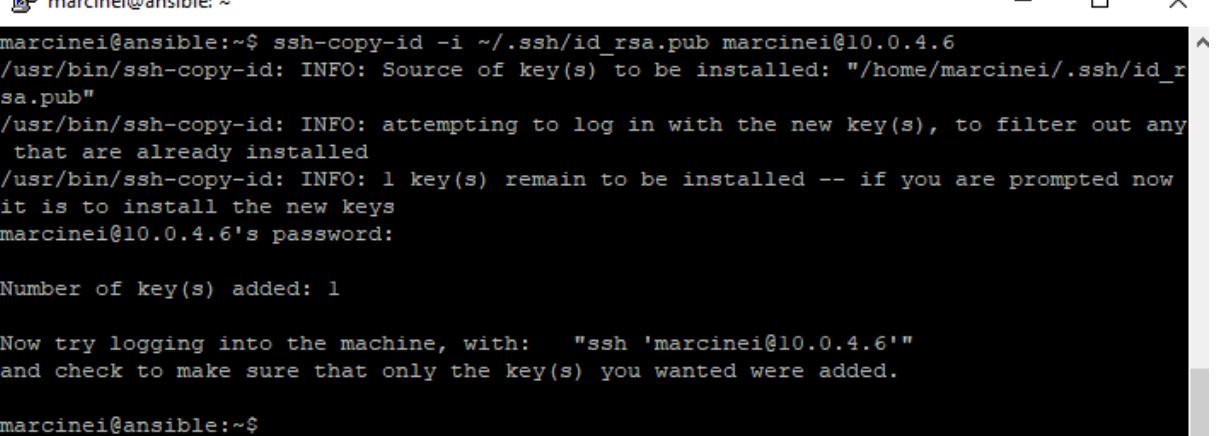
Figura 81 – Criação de chave SSH

Esta chave deve ser copiada para pasta onde o serviço de acesso remoto será informado desta informação, permitindo o login automático e seguro, segue comando para realizar a cópia.

```
cp -p ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

Também é preciso copiar a chave do usuário “Marcinei” do host “ansible” para o host “app”. Este procedimento é necessário pois o ansible utiliza a conexão SSH com os hosts e se a autenticação requerer login manual a playbook não será executada. Segue abaixo comando necessário e a captura de tela com a execução.

```
ssh-copy-id -i ~/.ssh/id_rsa.pub app@10.0.4.6
```



```

marcinei@ansible:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub marcinei@10.0.4.6
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/marcinei/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
marcinei@10.0.4.6's password:

Number of key(s) added: 1

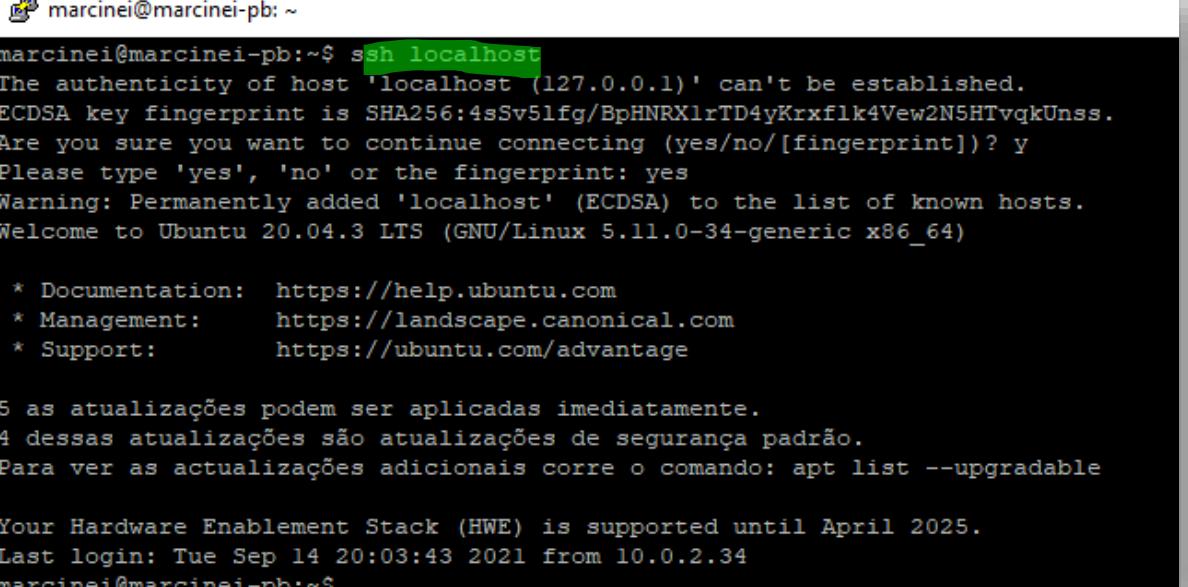
Now try logging into the machine, with:    "ssh 'marcinei@10.0.4.6'"
and check to make sure that only the key(s) you wanted were added.

marcinei@ansible:~$
```

Figura 82 – Cópia chave de acesso SSH

Para realizar o teste de conectividade de forma automática é necessário realizar o seguinte comando e em seguida confirmar o aceite desta configuração respondendo com “yes”, segue:

ssh localhost



```

marcinei@marcinei-pb:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:4sSv5lfg/BpHNRXlrTD4yKrxflk4Vew2N5HTvqkUnss.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-34-generic x86_64)

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

5 as atualizações podem ser aplicadas imediatamente.
4 dessas atualizações são atualizações de segurança padrão.
Para ver as actualizações adicionais corre o comando: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Sep 14 20:03:43 2021 from 10.0.2.34
marcinei@marcinei-pb:~$
```

Figura 83 – Teste de acesso SSH com login automático

A próxima etapa é a instalação do Ansible no host “ansible”, sendo necessário no primeiro passo adicionar ao sistema o repositório do software Ansible. O segundo comando

consiste em instalar o próprio software e por último a verificação da versão instalada. Seguem os comandos:

```
sudo apt-add-repository ppa:ansible/ansible
```

```
sudo apt install ansible
```

```
ansible --version
```

A atualização do ubuntu através do comando “*sudo apt update*” é indicada para permitir que a instalação utilize sempre a última versão dos softwares. Abaixo estão capturadas telas demonstrando a versão instalada e em seguida teste de funcionamento de um playbook teste.

Segue:

```
marcinei@ansible:~$ ansible --version
ansible [core 2.11.6]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['~/home/marcinei/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = ~/home/marcinei/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Sep 28 2021, 16:10:42) [GCC 9.3.0]
  jinja version = 2.10.1
  libyaml = True
marcinei@ansible:~$
```

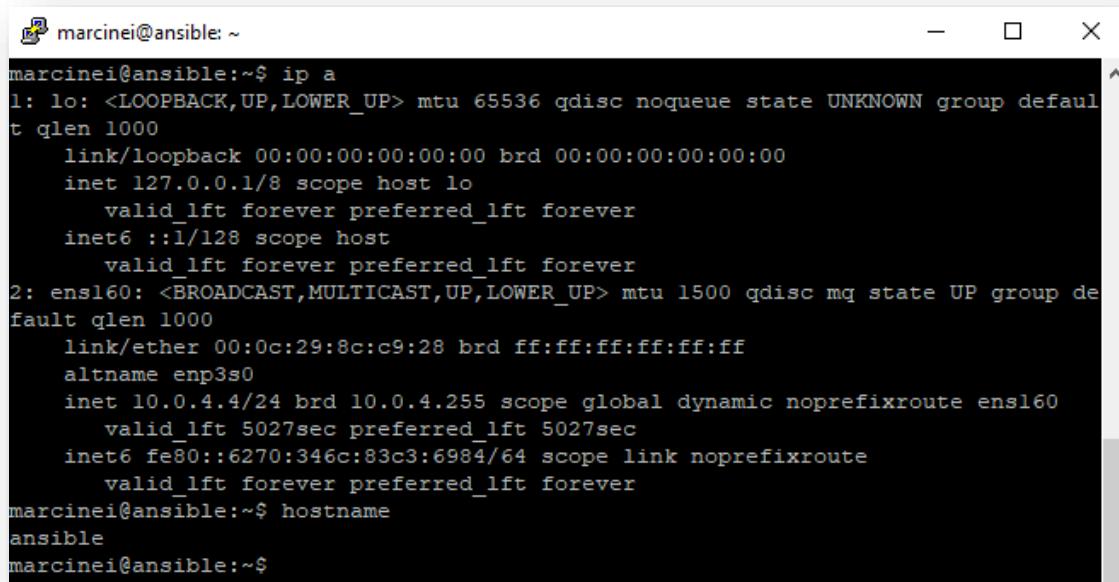
Figura 84 – Versão do Ansible instalado

Para permitir ao usuário criado durante a instalação executar comandos similares ao usuário **root** é preciso executar os seguintes comandos:

```
sudo su
```

```
echo "marcinei ALL=NOPASSWD: ALL" >> /etc/sudoers
```

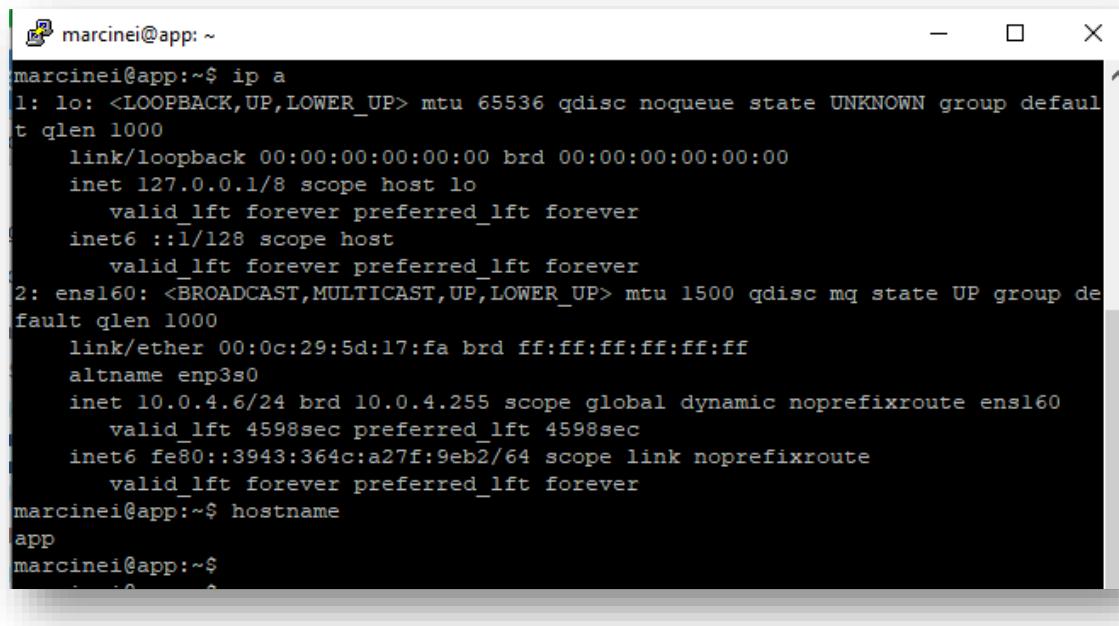
Nas próximas imagens estão demonstradas o acesso SSH as duas VMS Ubuntu, segue:



```

marcinei@ansible:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:8c:c9:28 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 10.0.4.4/24 brd 10.0.4.255 scope global dynamic noprefixroute ens160
        valid_lft 5027sec preferred_lft 5027sec
    inet6 fe80::6270:346c:83c3:6984/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
marcinei@ansible:~$ hostname
ansible
marcinei@ansible:~$
```

Figura 85 – Acesso SSH ao host “ansible”



```

marcinei@app:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:5d:17:fa brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 10.0.4.6/24 brd 10.0.4.255 scope global dynamic noprefixroute ens160
        valid_lft 4598sec preferred_lft 4598sec
    inet6 fe80::3943:364c:a27f:9eb2/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
marcinei@app:~$ hostname
app
marcinei@app:~$
```

Figura 86 – Acesso SSH ao host “app”

Nesta etapa forma realizadas configurações básicas nos dois hosts e também a implantação do ansible no host “ansible”. Na sequência a implantação do recurso Docker no host “app”.

7.7.6 Docker

O projeto desenvolvido tem como base fornecer um servidor Wordpress operacional, sendo que as primeiras etapas consistem em demonstrar apenas a implantação do Docker e pré requisitos no host “app” sendo executado na máquina virtual “UbuntuAPP”.

A primeira etapa será preparar o sistema operacional para a instalação do docker, começando pela criação de pasta específica para esta implantação:

```
mkdir wordpress-docker
cd wordpress-docker
```

Abaixo serão executados comandos instalando pacotes de suporte para o funcionamento do docker.

```
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    software-properties-common
```

O próximo comando realiza o download da chave de assinatura digital do docker, possibilitando autenticar os downloads futuros.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

No comando seguinte é exibido a identificação figerprint da chave adicionada.

```
sudo apt-key fingerprint 0EBFCD88
```

A próxima linha de comando adicionará ao repositório do Linux o endereço de downloads do próprio docker.

```
sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) \
    stable"
```

Nesta etapa será instalado realmente o docker.

```
sudo apt-get install docker-ce
```

Como pré requisitos para a execução do playbook do Ansible é necessário executar os seguintes comandos:

```
sudo apt install python3-pip
sudo pip3 install docker
```

Após estas etapas está concluído a etapa de preparação do ambiente, possibilitando a execução da automação da implantação através do código armazenado no portal Git.

7.7.7 Código Fonte

A instalação do Git é realizada com o comando abaixo. Segue também em seguida imagem com a execução do comando “git init” que define o diretório para a operação com o Git:

```
sudo apt install git
```

```
marcinei@marcinei-pb: ~/wordpress-ansible
marcinei@marcinei-pb:~/wordpress-ansible$ git init
Repositório vazio Git inicializado em /home/marcinei/.wordpress-ansible/.git/
```

Figura 87 – Inicialização do Git no diretório especificado

Para que as informações do git local sejam enviadas para o Github é necessário configurar a conta e o email cadastrados no portal. Seguem comandos:

```
git config --global user.email "marcinei.zimmermann@al.infnet.edu.br"
git config --global user.name "marcineizimmermann"
```

```

marcinei@ansible: ~/wordpress-ansible
marcinei@ansible:~/wordpress-ansible$ git config --global user.email "marcinei.zimmermann@al.infnet.edu.br"
marcinei@ansible:~/wordpress-ansible$ git config --global user.name "marcinei zimmermann"

```

Figura 88 – Configuração da conta do git

Para que a automação do versionamento dos arquivos estabeleça comunicação é preciso criar a conta no site oficial do github:

Github.com

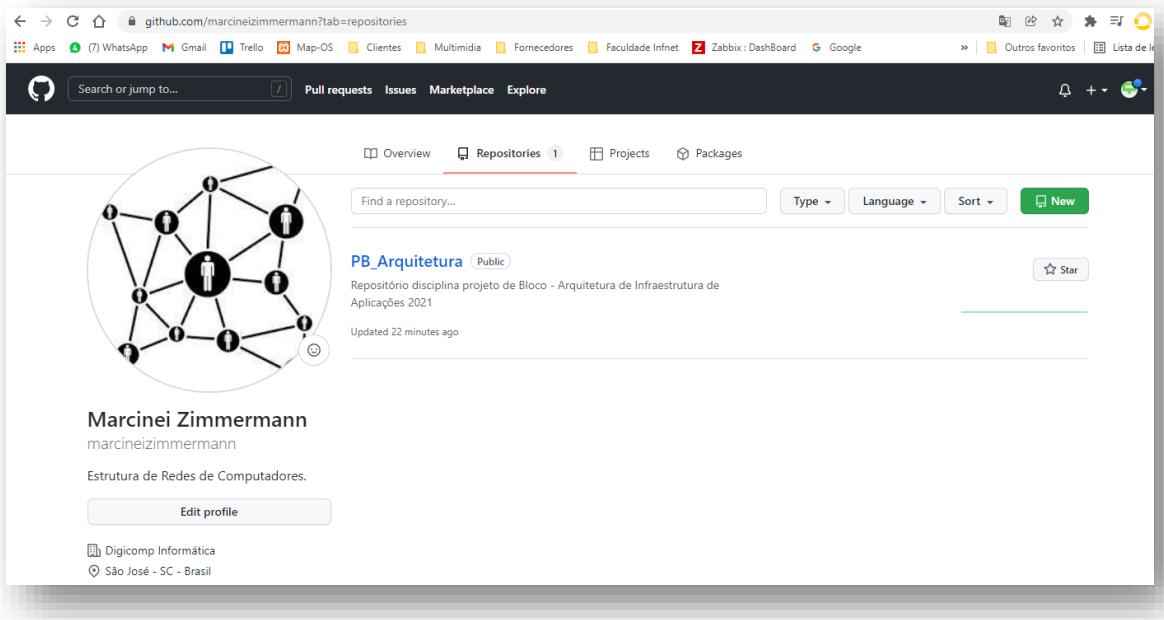


Figura 89 – Página Github Marcinei Zimmermann

Criado o acesso é necessário criar a chave token para que o git instalado no Ubuntu possa comunicar com o portal online, onde ficará armazenada de maneira definitiva e não requerendo autenticação nas próximas execuções e autenticações. Segue tela demonstrando a criação do token.

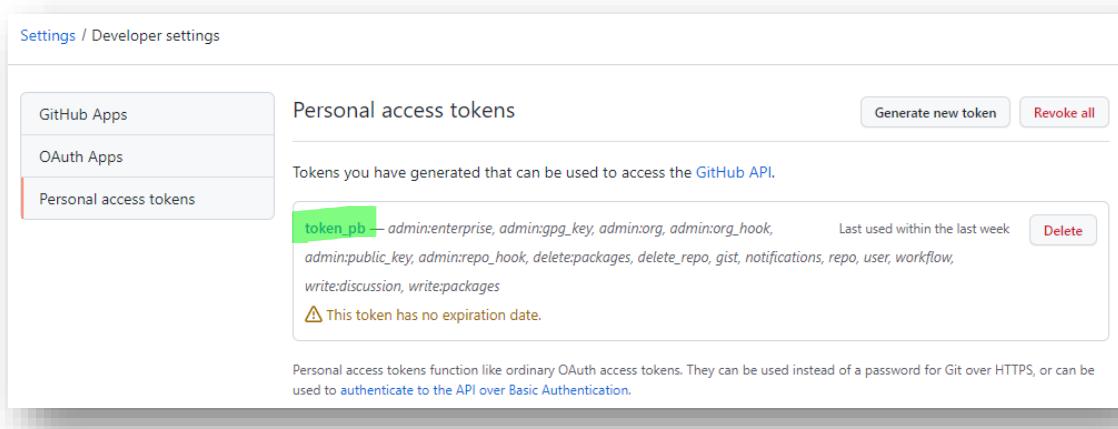
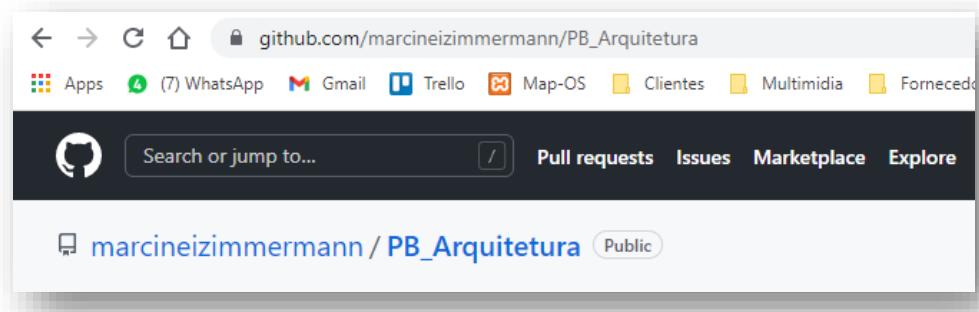


Figura 90 – Criação do token

Também será necessário criar o projeto que receberá as informações do git local, sendo este com o endereço:

https://github.com/marcineizimmermann/PB_Arquitetura

Segue tela com a captura da tela do site com o acesso a este projeto.



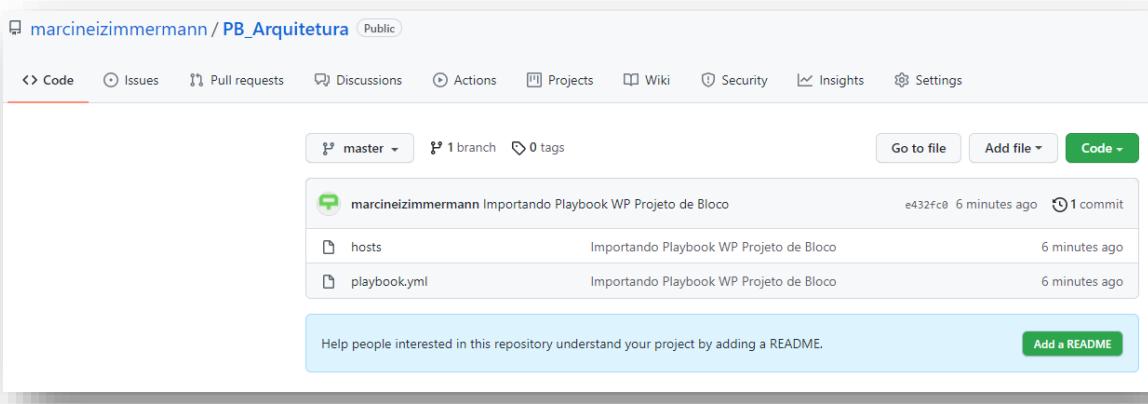


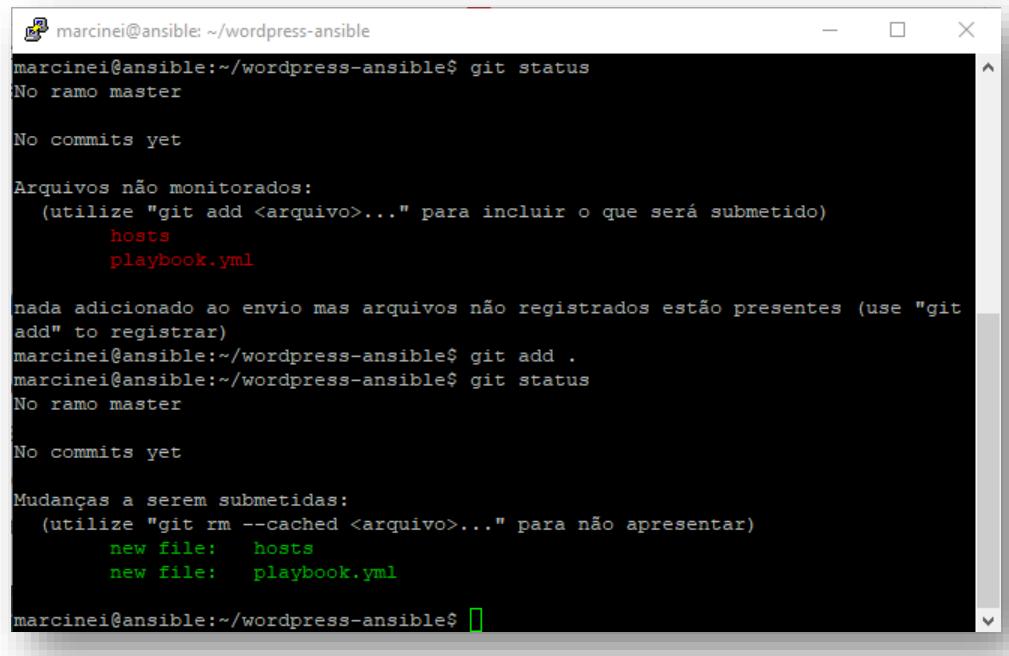
Figura 91 – Criação do Projeto referente ao projeto.

Através do terminal do host “ansible” é ideal que seja executado comando para verificar o estado do git. Segue comando:

```
git status
```

No comando abaixo é adicionado ao INDEX local do git, que funciona como uma área reservada onde os arquivos podem ser enviados para o portal GitHub. Com a utilização do asterístico é informado que deve ser copiado todo o conteúdo da pasta e sub pastas, enquanto que se for utilizado o ponto estará indicando a cópia do conteúdo da pasta onde está sendo executado o comando. Também é possível adicionar arquivos individualmente.

```
git add *
```



```

marcinei@ansible: ~/wordpress-ansible
marcinei@ansible:~/wordpress-ansible$ git status
No ramo master

No commits yet

Arquivos não monitorados:
  (utilize "git add <arquivo>..." para incluir o que será submetido)
    hosts
    playbook.yml

nada adicionado ao envio mas arquivos não registrados estão presentes (use "git
add" to registrar)
marcinei@ansible:~/wordpress-ansible$ git add .
marcinei@ansible:~/wordpress-ansible$ git status
No ramo master

No commits yet

Mudanças a serem submetidas:
  (utilize "git rm --cached <arquivo>..." para não apresentar)
    new file:   hosts
    new file:   playbook.yml

marcinei@ansible:~/wordpress-ansible$ 

```

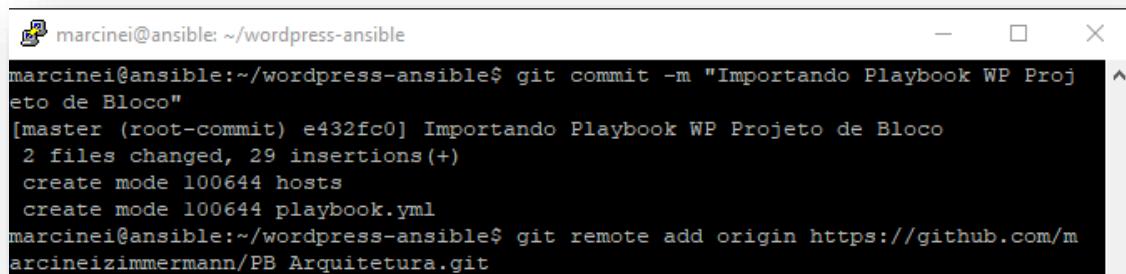
Figura 92 – Estado do git e adição de arquivos

Para confirmar as informações adicionadas ao git local é utilizado o seguinte comando, salientando que não está sendo publicadas as informações no portal.

```
git commit -m "Importando playbook WP Projeto de Bloco"
```

A próxima etapa é estabelecer conexão com o portal através do comando abaixo:

```
git remote add origin https://github.com/marcineizimmermann/PB_Arquitetura
```



```

marcinei@ansible: ~/wordpress-ansible
marcinei@ansible:~/wordpress-ansible$ git commit -m "Importando Playbook WP Proj
eto de Bloco"
[master (root-commit) e432fc0] Importando Playbook WP Projeto de Bloco
 2 files changed, 29 insertions(+)
  create mode 100644 hosts
  create mode 100644 playbook.yml
marcinei@ansible:~/wordpress-ansible$ git remote add origin https://github.com/m
arcineizimmermann/PB_Arquitetura.git

```

Figura 93– Commit e conexão com o Github

Desta forma é possível realizar o envio das informações para o repositório remoto. Segue comando e captura de tela demonstrando o êxito no envio.

```
git push -u origin master
```

```
marcinei@ansible: ~/wordpress-ansible
marcinei@ansible:~/wordpress-ansible$ git push -u origin master
Username for 'https://github.com': marcineizimmermann
Password for 'https://marcineizimmermann@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 568 bytes | 568.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/marcineizimmermann/PB_Arquitetura.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
marcinei@ansible:~/wordpress-ansible$
```

Figura 94 – Push git

Abaixo esta demonstrada captura de tela com o portal atualizado com as informações enviadas ao repositório remoto. Segue:

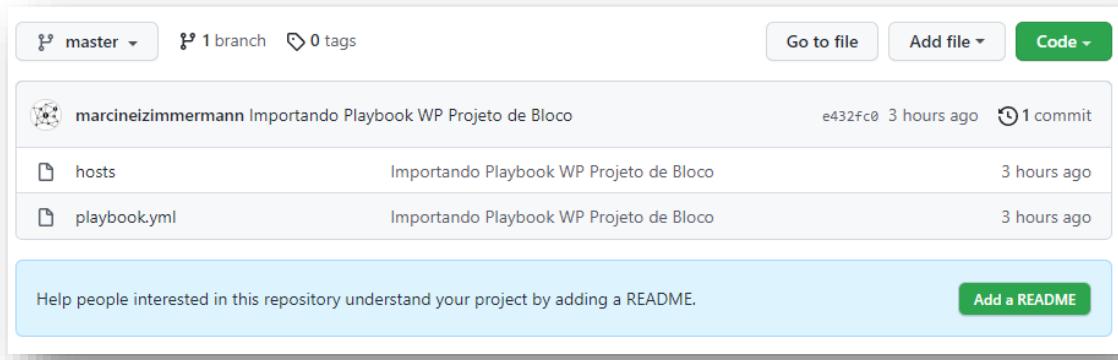


Figura 95 – Arquivos enviados para o Github

Segue abaixo link para acesso do repositório a ser utilizado pelo presente projeto:

https://github.com/marcineizimmermann/PB_Arquitetura

Na simulação não foi necessário realizar o download da informação armazenada no Github, mas em ambientes em que as implantações são rotineiras a necessidade de utilizar código modelo é fundamental e uma prática comum. Abaixo será relacionado comando para

clonar os arquivos no servidor Ansible, desta forma é possível reutilizar e adotar padrão de implantações.

```
git clone https://github.com/marcineizimmermann/PB_Arquitetura
```

Desta forma o ambiente de desenvolvimento é isolado da implantação, já que é possível adicionar a informação dos playbooks ao Github e posterior clonagem para o servidor Ansible. O controle de versionamento e alterações se torna fundamental para o controle das implantações e da linha de código a ser utilizada.

Após a instalação e configuração da ferramenta responsável por criar o repositório é possível apresentar o código utilizado para implantação. No projeto será aplicado a orquestração de criação de containers Docker para a aplicação WordPress.

O WordPress demanda a utilização de dois containers: Banco de dados MySQL 5.7 e o próprio WordPress. O Docker, o sistema operacional e a máquina virtual Linux Ubuntu serão pré instalados e configurados conforme etapas de implantação anteriores.

A criação da estrutura do ansible consiste inicialmente em definir a pasta onde será armazenado os arquivos e posteriormente conexão com o Github. A pasta definida é a seguinte:

```
/home/marcinei/wordpress-ansible
```

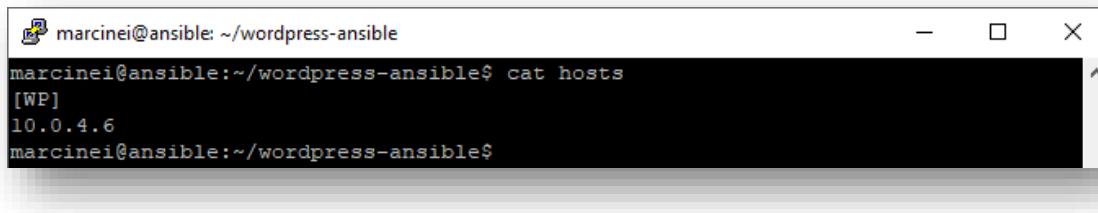
Dentro desta pasta é criado inicialmente o arquivo responsável por relacionar os hosts que serão orquestrados. Seguem os comandos necessários para criação do arquivo e inserção da informação, bem como o seu conteúdo:

```
touch hosts
```

```
nano hosts
```

```
[WP]
10.0.4.6
```

Segue captura de tela com o conteúdo do arquivo criado:



```

marcinei@ansible: ~/wordpress-ansible
marcinei@ansible:~/wordpress-ansible$ cat hosts
[WP]
10.0.4.6
marcinei@ansible:~/wordpress-ansible$
```

Figura 96 – Conteúdo arquivo hosts

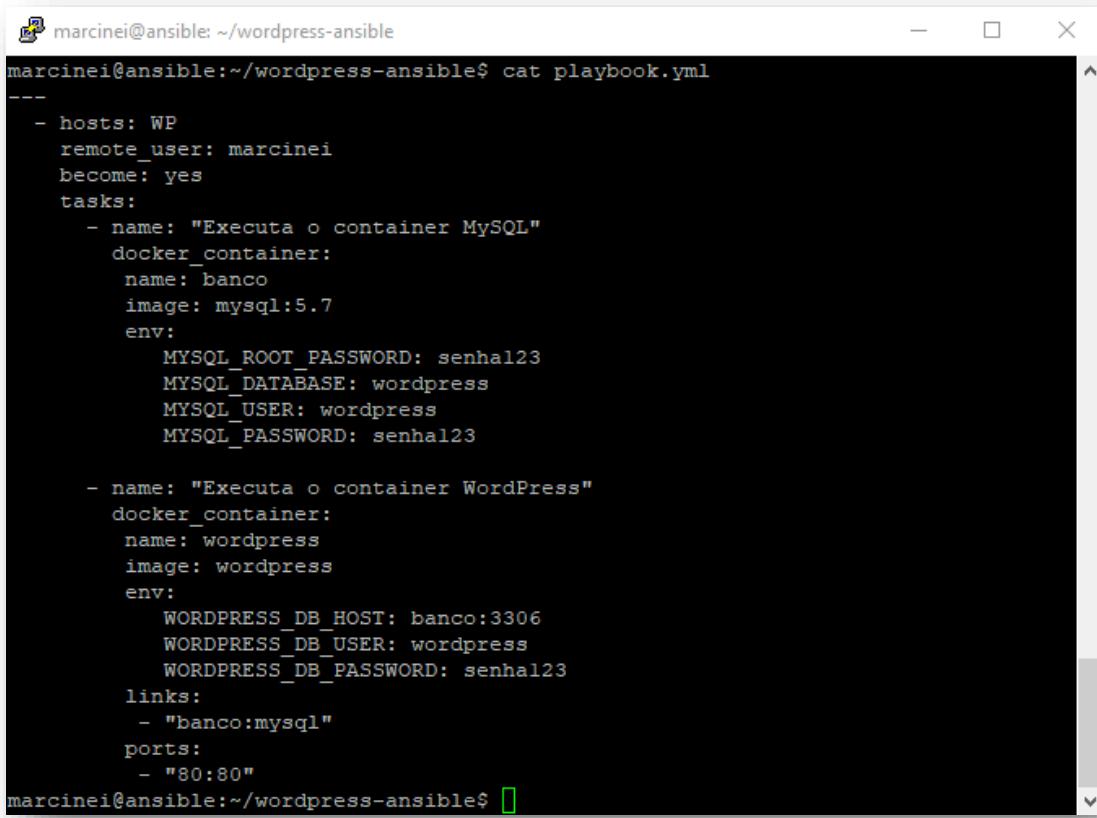
Para a criação do arquivo principal também será necessário criar o arquivo, editar e inserir as informações. Seguem os comandos necessários:

```
touch playbook.yml
```

```
nano playbook.yml
```

```
---
- hosts: WP
  remote_user: marcinei
  become: yes
  tasks:
    - name: "Executa o container MySQL"
      docker_container:
        name: banco
        image: mysql:5.7
        env:
          MYSQL_ROOT_PASSWORD: senha123
          MYSQL_DATABASE: wordpress
          MYSQL_USER: wordpress
          MYSQL_PASSWORD: senha123
    - name: "Executa o container WordPress"
      docker_container:
        name: wordpress
        image: wordpress
        env:
          WORDPRESS_DB_HOST: banco:3306
          WORDPRESS_DB_USER: wordpress
          WORDPRESS_DB_PASSWORD: senha123
        links:
          - "banco:mysql"
      ports:
        - "80:80"
```

Abaixo segue imagem com a captura da tela da edição do arquivo “playbook.yml”, segue:



```
marcinei@ansible: ~/wordpress-ansible
marcinei@ansible:~/wordpress-ansible$ cat playbook.yml
---
- hosts: WP
  remote_user: marcinei
  become: yes
  tasks:
    - name: "Executa o container MySQL"
      docker_container:
        name: banco
        image: mysql:5.7
      env:
        MYSQL_ROOT_PASSWORD: senhal23
        MYSQL_DATABASE: wordpress
        MYSQL_USER: wordpress
        MYSQL_PASSWORD: senhal23

    - name: "Executa o container WordPress"
      docker_container:
        name: wordpress
        image: wordpress
      env:
        WORDPRESS_DB_HOST: banco:3306
        WORDPRESS_DB_USER: wordpress
        WORDPRESS_DB_PASSWORD: senhal23
      links:
        - "banco:mysql"
      ports:
        - "80:80"
marcinei@ansible:~/wordpress-ansible$
```

Figura 97 – Conteúdo do arquivo “playbook.yml”

Desta forma fica demonstrado as configurações fundamentais para o funcionamento da aplicação. Na próxima etapa serão apresentadas configurações de personalização da aplicação para a empresa Digicomp.

7.7.8 Pós Configurações

Após a execução do playbook é necessário a personalização do portal, sendo realizado diretamente pelo gerenciador de conteúdo Web WordPress. Como exemplo de configuração inicial está demonstrado a utilização de um tema da página web. Segue imagem:

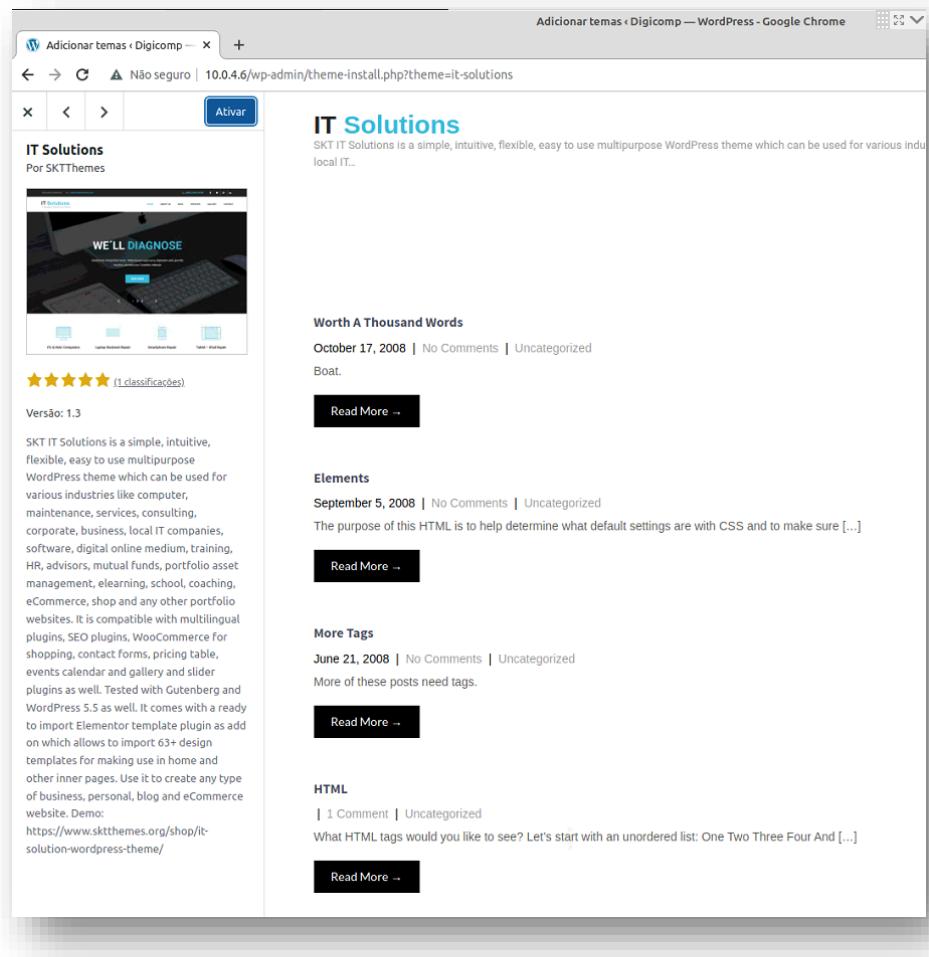


Figura 98 – Exemplo de personalização da página web

Abaixo está apresentado o menu de configurações do WordPress, onde são possíveis as configurações estéticas, de conteúdo, página, plugins, usuários, entre outros.



Figura 99 – Menu de configurações do WordPress

7.8 Testes

Após realizar todas as configurações é possível executar a implantação dos conteiners contendo o banco de dados e o WordPress. Através do terminal do host “ansible” deve ser executado o seguinte comando:

```
ansible-playbook -i hosts playbook.yml
```

Abaixo está capturado tela com a execução do playbook. Segue:

```
marcinei@ansible:~/wordpress-ansible
marcinei@ansible:~$ cd wordpress-ansible/
marcinei@ansible:~/wordpress-ansible$ ansible-playbook -i hosts playbook.yml

PLAY [WP] ****
TASK [Gathering Facts] ****
ok: [10.0.4.6]

TASK [Executa o container MySQL] ****
[DEPRECATION WARNING]: The container_default_behavior option will change its default value from "compatibility" to "no_defaults" in community.docker 2.0.0. To remove this warning, please specify an explicit value for it now. This feature will be removed from community.docker in version 2.0.0. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [10.0.4.6]

TASK [Executa o container WordPress] ****
changed: [10.0.4.6]

PLAY RECAP ****
10.0.4.6 : ok=3    changed=2    unreachable=0    failed=0    s
kipped=0   rescued=0   ignored=0

marcinei@ansible:~/wordpress-ansible$
```

Figura 100 – Execução correta do playbook

Segue imagem com status do Docker no host “app”, onde é possível constatar a criação e execução dos conteiners.

```

marcinei@app:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND      CREATED        STATUS          PORTS
 NAMES
2689d57025c6  wordpress "docker-entrypoint.s..."  3 hours ago   Up 3 hours    0.0.0.0:80->80/tcp
               wordpress
47076f08f69f  mysql:5.7 "docker-entrypoint.s..."  4 hours ago   Up 3 hours    3306/tcp, 33060/tcp
               banco
marcinei@app:~$
```

Figura 101 – Conteiners em execução no host “app”

Para realizar o teste de funcionamento do WordPress é preciso inserir o endereço IP do servidor em um navegador web. Na primeira imagem o acesso é realizado no próprio host “app” onde está sendo executado o gerenciador Web, sendo inserido o endereço “127.0.0.1” que significa “localhost”, ou seja, o endereço do próprio host. Segue imagem.

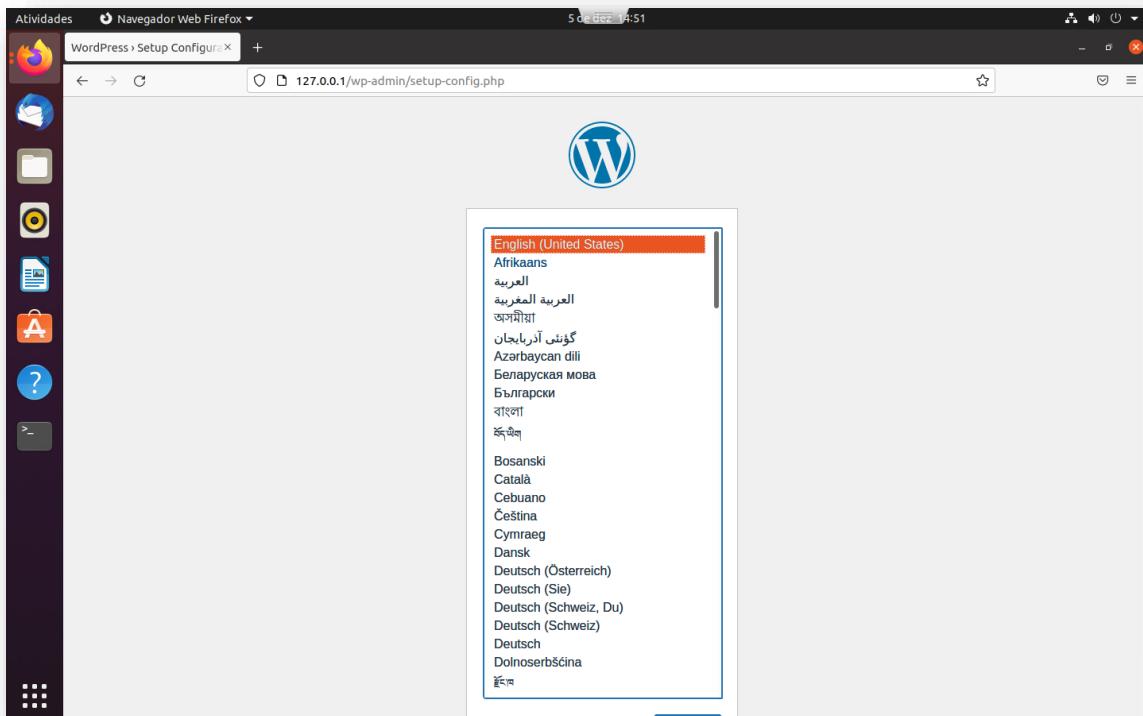


Figura 102 – Acesso ao WordPress no próprio host “app”

Na sequência é demonstrado o acesso em outro host configurado na mesma rede do WordPress. Segue imagem:

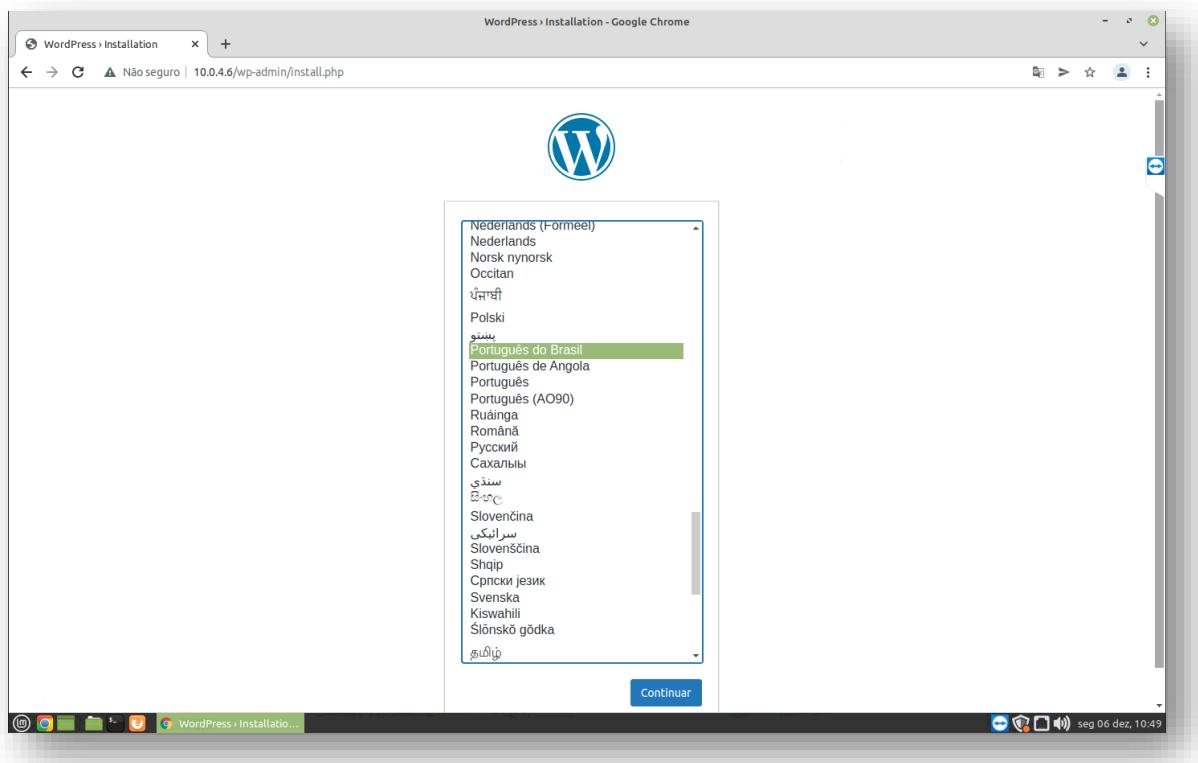


Figura 103 – Acesso ao WordPress através de host na mesma rede

Após selecionar o idioma, conforme imagem acima, a configuração inicial do WordPress solicita as seguintes informações: Título do Site, Nome do Usuário, Senha e Email. Segue imagem abaixo:

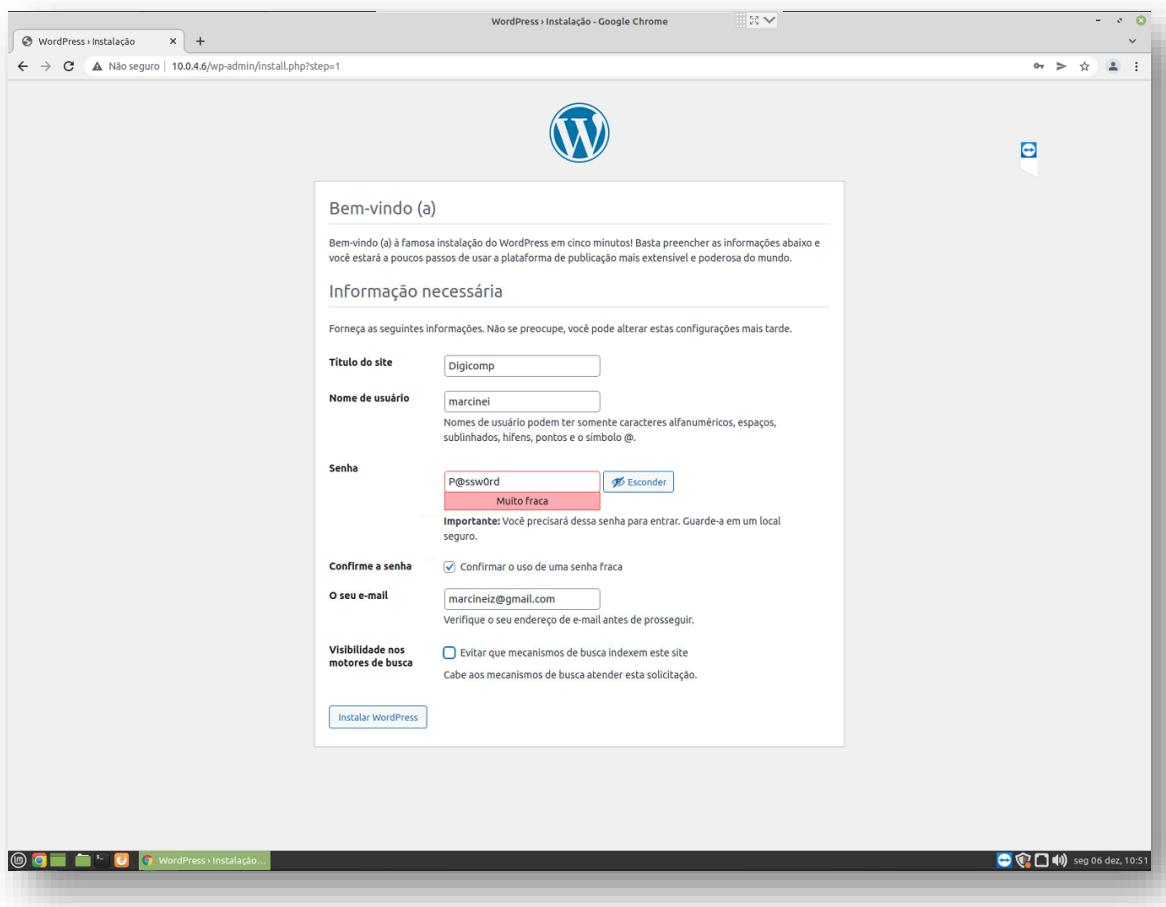


Figura 104 – Tela de configuração inicial do WordPress

Por fim, é apresentado a tela com a finalização da instalação e configuração do gerenciador de conteúdo Web. A partir desta etapa é possível personalizar o portal.

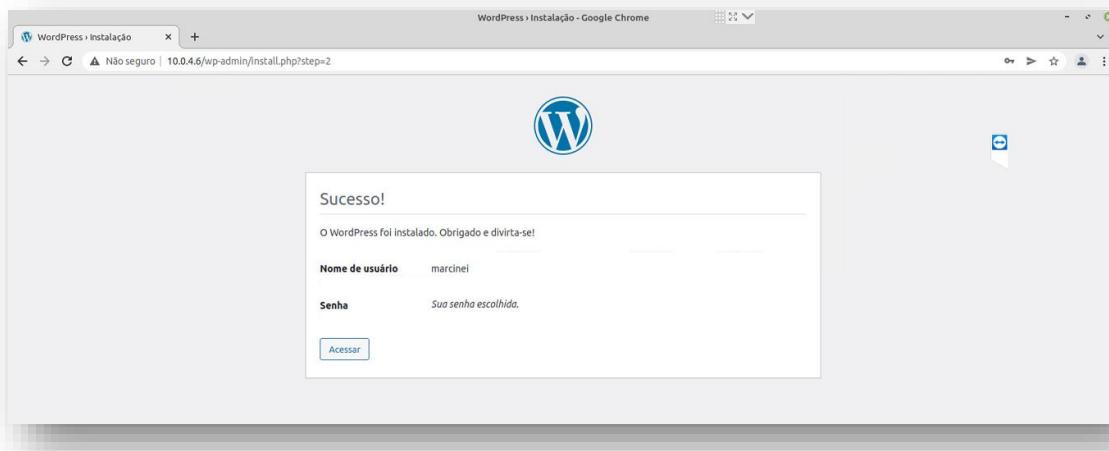


Figura 105 – Conclusão da instalação e configuração

Na próxima tela é apresentado a tela de login do WordPress. Segue imagem:

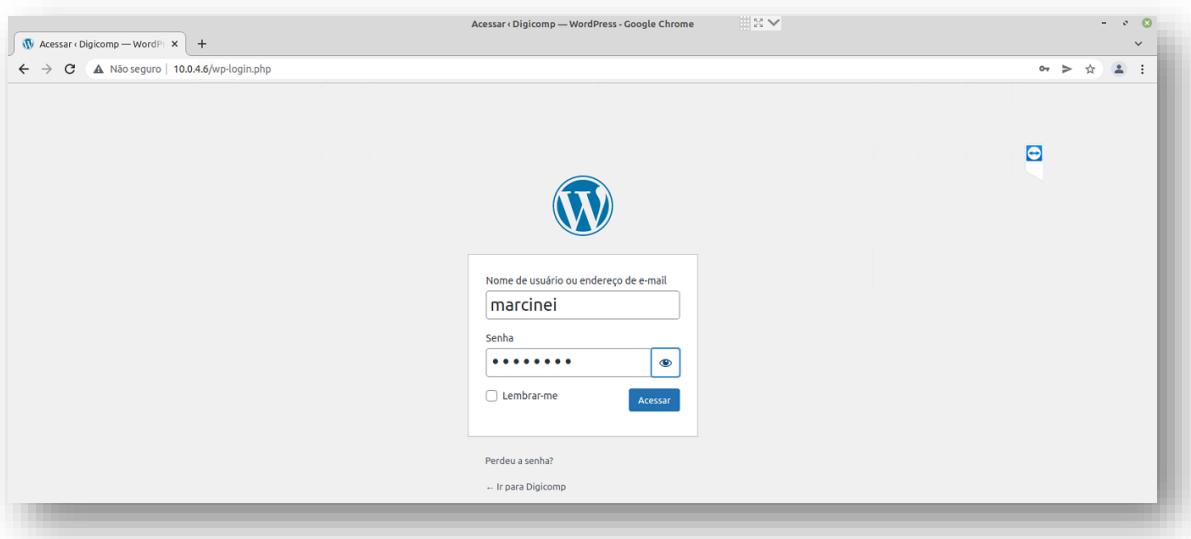


Figura 106 – Tela de login do WordPress

Na próxima imagem é apresentado a tela de gerenciamento (wp-admin) do WordPress, sendo possível realizar todas as configurações, ajustes, adição de plugins e personalizações do portal Web.

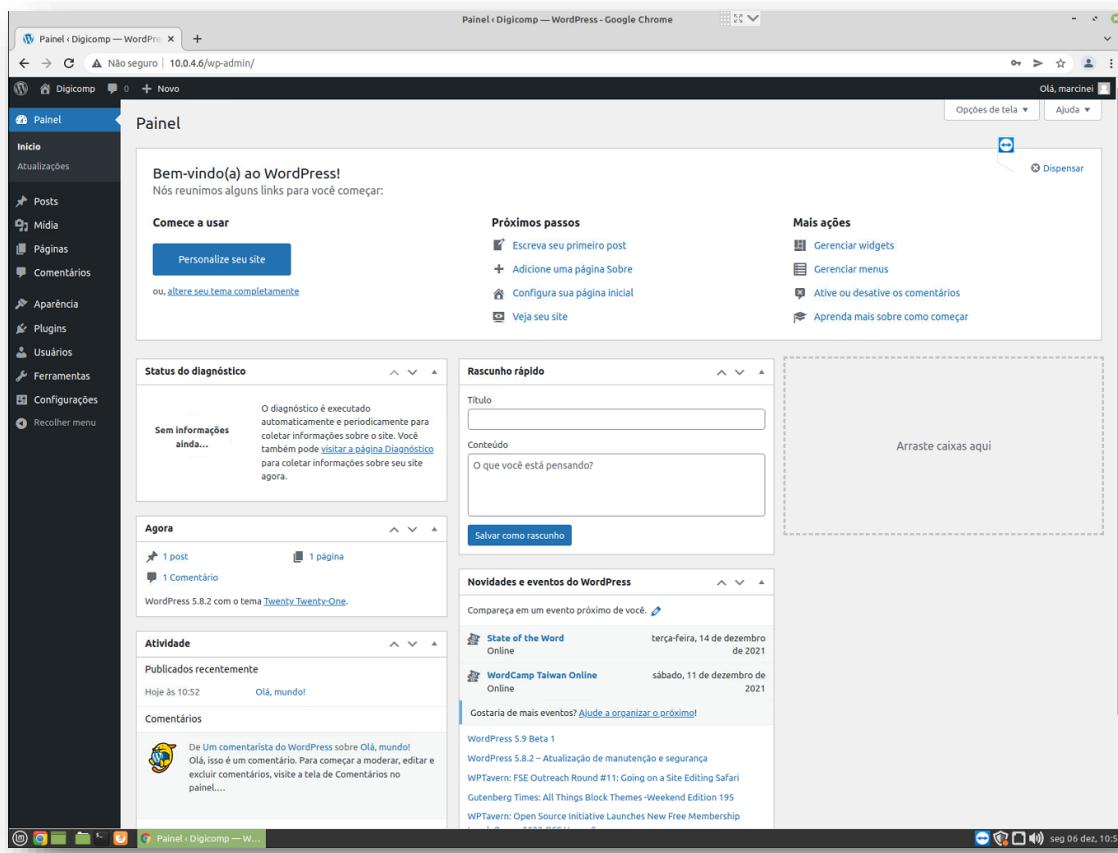


Figura 107 – Portal de administração do WordPress

Abaixo está apresentado a página da Digicomp com as primeiras informações, possibilitando a criação de menus, ferramentas e utilitários a serem implantados. Como o foco deste projeto é poder disponibilizar a aplicação, desta maneira fica completo o desenvolvimento e implantação.



Figura 108 – Página Web da empresa Digicomp

O projeto é o início, portanto, o primeiro passo para a aplicação ser executada e utilizada pela estrutura está completa.

8 Análise da Estrutura

Ao constatar o funcionamento da aplicação é determinante a necessidade de analisar os aspectos que apresentaram falha e que demandam atenção, principalmente pelo fato de que o portal web será utilizado pela empresa e terá participação fundamental na sua operação.

O primeiro ponto a ser verificado e que demonstrou inconsistência durante a execução é a execução da playbook, onde em vários momentos a criação do contêiner do banco de dados MySQL apresentava falha, mas ao repetir o procedimento a instalação era concluída e com êxito.

A programação em yaml também apresenta aspectos delicados de tabulação em sua linha de código, o que demanda atenção e foco para encontrar a formatação correta.

Também ficou evidente a atenção por buscar sempre atualizações dos sistemas operacionais e aplicações, pois em vários momentos da criação do ambiente alguns recursos não funcionaram, após atualização do sistema estes mesmos voltaram a funcionar normalmente.

Em relação ao funcionamento da estrutura, ficou nítido que a estrutura necessária para a execução demanda maior capacidade de hardware, visto que em vários momentos a execução de tarefas apresentavam lentidão e demora excessiva.

A redundância e a fragilidade no armazenamento é talvez o ponto mais temeroso, pois qualquer problema técnico no equipamento provocaria parada total do laboratório e por consequência atrasos.

Fica evidente após esta implementação, que as demandas por recursos tecnológicos de processamento, armazenamento, conectividade, gerenciamento e softwares, devem ser priorizadas. Com o aumento da utilização do portal é certo de que em pouco tempo a estrutura atual da empresa exigirá maior quantidade de recursos. A questão financeira para a continuidade do projeto deve ter atenção especial pela administração, senão correrá risco de que a aplicação se torne um problema e não uma solução.

Apesar da utilização de softwares livres, ficou claro a necessidade de profissionais com conhecimento específico nas aplicações. Os ajustes e correções do ambiente serão rotineiros e demandaram atenção, tornando assim mais um ponto a ser considerado pela administração da empresa.

9 Conclusão

Após demonstrar os aspectos relacionados ao desenvolvimento do projeto, é possível determinar a importância e utilidade de se utilizar ambientes com automação de implantações. O profissional Devops tem papel importante e até fundamental para que as aplicações sejam disponibilizadas de maneira eficiente e ágil, como a maioria delas demanda.

A utilização do ansible possibilitou de maneira simplificada e organizada a implantação de soluções para a empresa. As possibilidades de aplicação desta ferramenta são tão vastas

que a sua utilização é imprescindível para que os profissionais técnicos consigam atender a crescente demanda e crescente agilidade no fornecimento de tecnologias.

O projeto possibilitou claramente o desenvolvimento de conhecimentos sobre aplicações, suas camadas, recursos necessários e suas estruturas de funcionamento. A metodologia de criação de um projeto exige que a metodologia utilizada seja ampla e contemplando vários aspectos, entre eles: hardware, virtualização, sistema operacional, recursos de processamento, armazenamento, conectividade, segurança e acima de tudo de organização.

O objetivo de alcançar nova visão sobre a implantação de uma aplicação foi alcançada, criando uma visão ampla dos aspectos que são necessários e que demandam atenção. Os pontos de vista do desenvolvedor e da equipe técnica não devem ser desconsiderados, o contrário, devem ser utilizados para criar um ambiente produtivo e de cooperação.

A preocupação administrativa e financeira sempre esteve presente no desenvolvimento do projeto, aspecto que por muitas vezes não foi levado em consideração, mas de grande prioridade para o aceite e execução do objetivo do projeto. Fundamental entender que as ideias e soluções só serão válidas se elas forem compatíveis e úteis para empresa.

Link Apresentação

A apresentação está disponível no seguinte link:

- https://drive.google.com/file/d/1laMiJeGqu3xwlBz1NuWVRHmQTzzMX_qQ/view?usp=sharing

Link Github

Segue endereço da página do Github com o projeto:

- https://github.com/marcineizimmermann/PB_Arquitetura

Referências Bibliográficas:

- 4 LINUX. O que é DevOps. < <https://4linux.com.br/o-que-e-devops/> >. Acesso em 17 ago. 2021.
- ANSIBLE. Ansible Documentation. 2021. Disponível em < <https://docs.ansible.com/ansible/latest/> >. Acesso em 04 dez. 2021.
- HAPPYCODERS. Ansible Tutorial: Setup Docker, MySQL, and WordPress with Ansible [Update 2020]. 2020. Disponível em < <https://www.happycoders.eu/devops/ansible-tutorial-setup-docker-mysql-wordpress/> >. Acesso em 06 dez. 2021.
- ICONECTADO WEB. Curso de WordPress 2021. 2019. Disponível em < <https://iconectado.com.br/curso-de-wordpress/> >. Acesso em 17 ago. 2021.
- IFSULDEMINAS. Clusters de alta disponibilidade em servidores Web utilizando sistemas livres. Disponível em < <https://jornada.ifsuldeminas.edu.br/index.php/jcmch1/jcmch1/paper/viewFile/1717/1196> >. Acesso em 23 nov. 2021.
- INFNET. Projeto de Bloco: Arquitetura de Infraestrutura de Aplicações. 2021. Disponível em < <https://lms.infnet.edu.br/moodle/course/view.php?id=4495> >. Acesso em 17 ago. 2021.
- NEOMIND. Processos: o que é? 2018. Disponível em < <https://www.neomind.com.br/blog/mapeamento-de-processos-as-is-to-be/> >. Acesso em 28 nov. 2021.
- ONLINEYAMLTOOLS. Online Yaml Validator. Disponível em < <https://onlineyamltools.com/validate-yaml> >. Acesso em 06 dez. 2021.
- REDHAT. Automação. Noções básicas do Ansible. Disponível em < <https://www.redhat.com/pt-br/topics/automation/learning-ansible-tutorial> >. Acesso em 17 ago. 2021.
- STUDIO VISUAL. Como criar uma página no WordPress Passo a Passo. Disponível em < <https://studiovvisual.com.br/wordpress/desenvolvimento/como-criar-uma-pagina-no-wordpress> >. Acesso em 06 dez. 2021.
- VMWARE. Compare edições do VMware vSphere. 2018. Disponível em < <https://www.vmware.com/content/dam/digitalmarketing/vmware/pt/pdf/vsphere/vmw-flyr-comparevsphereeditions-uslet.pdf> >. Acesso em 25 nov. 2021.
- VMWARE. Oracle Databases on VMware High Availability. 2011. Disponível em < <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/solutions/oracle/oracle-databases-on-vmware-high-availability-guidelines.pdf> >. Acesso em 25 nov. 2021.

- VMWARE. vCenter Server e Gerenciamento de Host. 2020. Disponível em < <https://docs.vmware.com/br/VMware-vSphere/7.0/vsphere-esxi-vcenter-server-701-host-management-guide.pdf> >. Acesso em 25 nov. 2021.
- VMWARE. vSphere High Availability. 2021. Disponível em < <https://www.vmware.com/br/products/vsphere/high-availability.html> >. Acesso em 25 nov. 2021.