



Instituto Infnet

GRADUAÇÃO EM REDE DE COMPUTADORES

ALUNO: CLAUDIO LUIS AZEVEDO GRANDE

PROFESSOR: FABIANO ALVES GISBERT

**FUNDAMENTOS DA ARQUITETURA DE INFRAESTRUTURA DE APLICAÇÕES
APRESENTAÇÃO**

30/11/2021

Índice

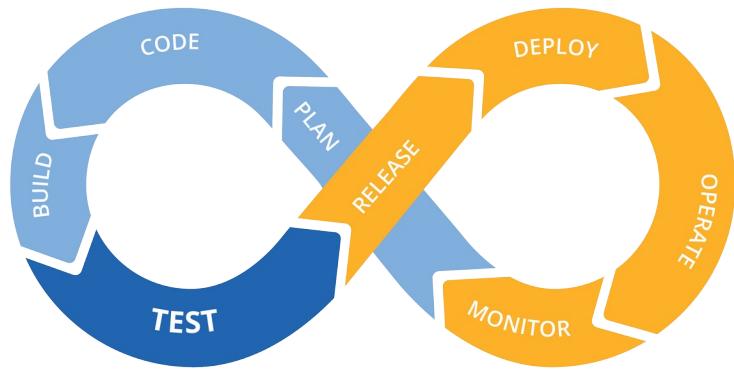
Sobre a CAZGR.....	3
Sobre a LOGICOL.....	3
Aplicação Ponto Remoto.....	4
Pilha de softwares da Aplicação.....	5
Projeto de infraestrutura de arquitetura.....	6
Soluções propostas de arquitetura da Infraestrutura.....	7
Solução OpenStack.....	7
Solução vSphere.....	8
Infraestrutura Ágil.....	9
Infraestrutura - To Be.....	11
Diagrama da infraestrutura Física e Lógica - To Be.....	13
Detalhes da infraestrutura Física e Lógica - To Be.....	14
Infraestrutura - As Is.....	15
Diagrama da infraestrutura Física Simulada - As Is.....	17
Processo de instalação do host VMware ESXi.....	18
Processo de instalação do FreeNAS.....	23
Processo de Configuração dos Volumes no FreeNAS.....	30
Processo de adição de um Datastore NAS no ESXi.....	35
Processo de criação das VMs usando Datastore Virtual NAS.....	38
Informações Gerais das VMs já instaladas “BDWAPP” e “ANSIBLE”.....	42
Instalação do Servidor ANSIBLE.....	43
Instalação do Servidor SSH.....	44
Instalação do Ansible.....	46
Instalação do Servidor BDWAPP.....	47
Preparando o ambiente Docker.....	48
Preparando um Playbook para containers.....	50
Escrevendo o arquivo de inventário (hosts).....	50
Criando o Playbook.....	51
Validando o Playbook.....	52
Acessando o WordPress pelo navegador.....	53
Preparando o ambiente GIT local.....	54
Fazendo um Commit.....	55
Fazendo um Push.....	55
Estimativa de Custos.....	57
Cronograma.....	58
Referências Bibliográficas.....	60

Sobre a CAZGR

A CAZGR é uma empresa TI que provê soluções inovadoras e serviços profissionais especialistas para fazer a ponte entre o presente e o futuro.

Oferecemos um conjunto de soluções de governança e tecnologia para uso máximo de todos os recursos da virtualização e cloud.

Nosso time de especialistas utiliza a cultura DevOps para fornecer valor continuamente ao cliente, através de práticas que integram o trabalho das equipes de desenvolvimento de sistemas e das equipes de operações de forma automatizada. Auxiliando nossos clientes no desafio constante de entregar novas soluções e, ao mesmo tempo, garantir a estabilidade do ambiente.



Sobre a LOGICOL

A LOGICOL é uma empresa de logística com mais de 10 anos de experiência no mercado, atua nos principais segmentos da cadeia logística, como transporte, armazenagem e distribuição. Com foco em pequenos e médios volumes.

Devido ao advento da pandemia, os colaboradores da empresa LOGICOL passaram a trabalhar remotamente e com isso houve a necessidade de implantar uma aplicação para registrar o ponto remotamente.

Aplicação Ponto Remoto

A aplicação de ponto remoto foi desenvolvida pela CAZGR, para ser utilizada através de uma plataforma web (Wordpress) que automatiza o envio e visualização de informações do Departamento Pessoal e RH, conectando empresa, gestores e colaboradores.

Esta aplicação tem o objetivo de dispensar o uso do REP (o aparelho de ponto eletrônico convencional). É uma aplicação simples, mas de vital importância para a empresa, já que controlará toda a gestão de ponto dos funcionários em cumprimento a todos os requisitos da portaria 373/2011 do ministério do trabalho.

O Ponto Remoto foi planejado para trabalhar em um infraestrutura ágil com práticas DevOps, usando Docker, GIT e Ansible. Com o objetivo de gerar valor ao cliente, trazendo agilidade, inovação e segurança.

Devido a natureza da aplicação e de sua importância, duas premissas são imperativas para o projeto de infraestrutura:

1. Tolerância a falhas e Alta disponibilidade
2. Segurança

Após estudos de viabilidade, a diretoria da empresa chegou a um consenso que toda infraestrutura para hospedar a sua aplicação deverá ser feita em um ambiente **on-primeses**. A LOGICOL tomou esta decisão principalmente por não querer aumentar seus custos com despesas fixas de operação (OPEX), mas sim aproveitar este momento de necessidade para se capitalizar, utilizando as despesas das aquisições desses novos ativos em investimento. Entretanto, os recursos financeiros são limitados devido ao atual cenário da economia.

Pilha de softwares da Aplicação

Para que esta aplicação funcione como foi planejada, alguns softwares são necessários.

São eles:

- Linux Ubuntu Server 20.04 LTS;
- PHP 7.4 ou superior;
- MySQL 5.6 ou superior;
- Apache;
- Docker

Para poder suportar todos esses requisitos de software, as seguintes configurações de hardware são necessárias:

- Cpu: 4 núcleos;
- Memória RAM: 64 GB;
- Armazenamento: 20TB de espaço;

Projeto de arquitetura de infraestrutura

Como sabemos, os colaboradores da empresa LOGICOL passaram a trabalhar remotamente e com isso houve a necessidade de implantar uma aplicação para registrar o ponto remotamente. Devido a natureza da aplicação, onde a necessidade de segurança, tolerância a falhas e alta disponibilidade são essenciais. Estamos aproveitando esta oportunidade para implantar uma solução de SDDC (data center definido por software).

O grande ganho do SDDC é a simplificação do provisionamento de aplicativos e recursos de TI para disponibilizá-los para uso em tempos muito menores do que em um data center tradicional. Tudo isso devido a forma como ele pode ser estruturado e controlado, pois como os recursos são gerenciados por software, as conexões físicas e os componentes de hardware não precisam mais ser fisicamente manipulados para implantar mudanças. Além disso, não menos importantes: redução de despesas, redução de complexidade e viabilização de um controle melhor da infraestrutura.

Primeiro de tudo, vamos fazer uma avaliação de todos os componentes que formam a arquitetura atual. Isso inclui hardwares (switches, servidores e storages) e softwares de implantação lógica para agregar conectividade e intercomunicação.

O próximo passo envolve a escolha de um software para gerenciar o SDDC com eficiência. Depois, já com a capacidade de processamento definida, lógica de rede configurada e software capaz de atender ao modelo proposto, a empresa já estará apta a usar esse novo modelo de data center.

Toda documentação e código de automação deste projeto de arquitetura de infraestrutura, poderá ser acessada através do link do repositório do GitHub:

<https://github.com/cazgr/WP-Docker-Ansible>

Soluções propostas de arquitetura da Infraestrutura

Diante das premissas apresentadas e o estudo de viabilidade feito pela LOGICOL, coube a nós da CAZGR apresentar dois cenários de arquitetura que se enquadram dentro destas bases pré estabelecidas.

Solução OpenStack

Openstack é uma plataforma de software para cloud computing livre e opensource que consiste de uma série de diferentes projetos responsáveis por desenvolver subsistemas separados, que juntos formam a solução OpenStack.

Ao implementarmos o OpenStack podemos dizer que estamos quase que construindo uma infraestrutura "de nível AWS", mas com a intenção de rodar apenas um único cliente dentro dela, a LOGICOL. Do ponto de vista de esforço de gerenciamento, suportar apenas um cliente, com uma única aplicação rodando em tantos servidores, não vale a pena. Além disso, o investimento inicial seria muito mais alto que a segunda solução a ser apresentada, a vSphere.

Dito isso, apesar de ser uma solução de alto nível, para a realidade e a necessidade do cliente, não é vantajoso a solução openstack. É como “matar uma mosca com uma bala de canhão”.

Solução vSphere

Como solução de SDDC (data center definido por software), estaremos utilizando os softwares da VMware que reúne o melhor processamento, armazenamento e sistema de rede virtualizados com um gerenciamento de nuvem abrangente. Este ambiente operacional comum permite executar, gerenciar, conectar e proteger rapidamente os aplicativos.

O vSphere é a plataforma de virtualização da Vmware e também base para uma infraestrutura baseada em nuvem. Os dois componentes principais do vSphere são:

ESXi

O ESXi é a plataforma de virtualização que será utilizada neste projeto. Ele simplifica a TI separando aplicativos e sistemas operacionais do hardware subjacente, fornecendo a base para criar e gerenciar uma infra-estrutura de TI virtualizada.

vCenter

O VMware vCenter Server é o hub universal para gerenciar o ambiente VMware vSphere. O vCenter Server fornece gerenciamento unificado de todos os hosts e máquinas virtuais de um data center a partir de um único console.

Além destes componentes base, outros componentes importantes farão parte da solução.

vMotion

Possibilita a migração em tempo real de máquinas virtuais entre servidores, vCenter Servers e para longas distâncias (até 100 milissegundos de tempo de ida e volta), sem interrupção para usuários ou perda de serviço, eliminando a necessidade de programar tempo de inatividade de aplicativos para manutenção planejada do servidor.

High-Availability (HA)

Oferece disponibilidade para aplicativos executados em máquinas virtuais independentes do sistema operacional para reduzir automaticamente o tempo de inatividade dos aplicativos. O High Availability fornece proteção de failover uniforme e econômica contra interrupções de hardware e sistema operacional em seu ambiente de TI virtualizado.

Distributed Resource Scheduler (DRS)

Fornece balanceamento de carga e agendamento em um cluster vSphere, garantindo que as máquinas virtuais em execução dentro do seu ambiente vSphere tenham os recursos de que precisam para funcionar de maneira eficaz e eficiente.

vSphere Data Protection (VDP)

É uma solução de backup e recuperação robusta, simples de implantar e baseada em disco. O VDP é totalmente integrado ao VMware vCenter Server e ao VMware vSphere Web Client. O VDP permite o gerenciamento centralizado e eficiente de tarefas de backup enquanto armazena backups em armazenamento de destino desduplicado.

Infraestrutura Ágil

Além da solução VMware vSphere, O Ponto Remoto foi planejado para trabalhar em um infraestrutura ágil com práticas DevOps, usando Docker, GIT e Ansible. Com o objetivo de gerar valor ao cliente, trazendo agilidade, inovação e segurança.

Docker

É uma plataforma de software que permite a criação, o teste e a implantação de aplicações rapidamente. O Docker cria pacotes de software em unidades padronizadas chamadas de contêineres que têm tudo o que o software precisa para ser executado, inclusive bibliotecas, ferramentas de sistema, código e runtime. Ao usar o Docker, é

possível implantar e escalar rapidamente aplicações em qualquer ambiente e ter a certeza de que o seu código será executado.

GIT e Ansible (Sistema de Versionamento e Automação)

Para este projeto estaremos usando também um sistema de controle de versão (GIT) para registrar alterações nos arquivos ao longo do tempo. Possibilitando reverter arquivos a um estado anterior, comparar as mudanças ao longo do tempo, além de tornar o código totalmente auditável.

O projeto a ser desenvolvido como dito acima, estará dentro de um processo de entrega contínua aonde o código ficará dentro de um ambiente compartilhado e estruturado (GIT) para ser implantado de forma automatizada (através do Ansible).

O Ansible é um mecanismo de automação de TI simples. Ele lida com gerenciamento de configuração, implantação de aplicativo, provisionamento em nuvem, execução de tarefas, automação de rede e muitas outras. Com essa automação, tarefas complexas e manuais são resumidas em simples cliques, agilizando assim todo o processo.

O Ansible é basicamente o que ligará o mundo operacional e o de desenvolvimento, ele usa a linguagem YAML, na forma de Ansible Playbooks que permite descrever os trabalhos de automação de uma forma extremamente simples.

Infraestrutura - To Be

Baseado nas premissas levantadas para o projeto de infraestrutura e de acordo com os requisitos de softwares que compõem a plataforma que suportará a aplicação, verificamos que será necessário para sustentar toda esta estrutura e ainda termos uma folga para possíveis ajustes e upgrades, os seguintes requisitos:

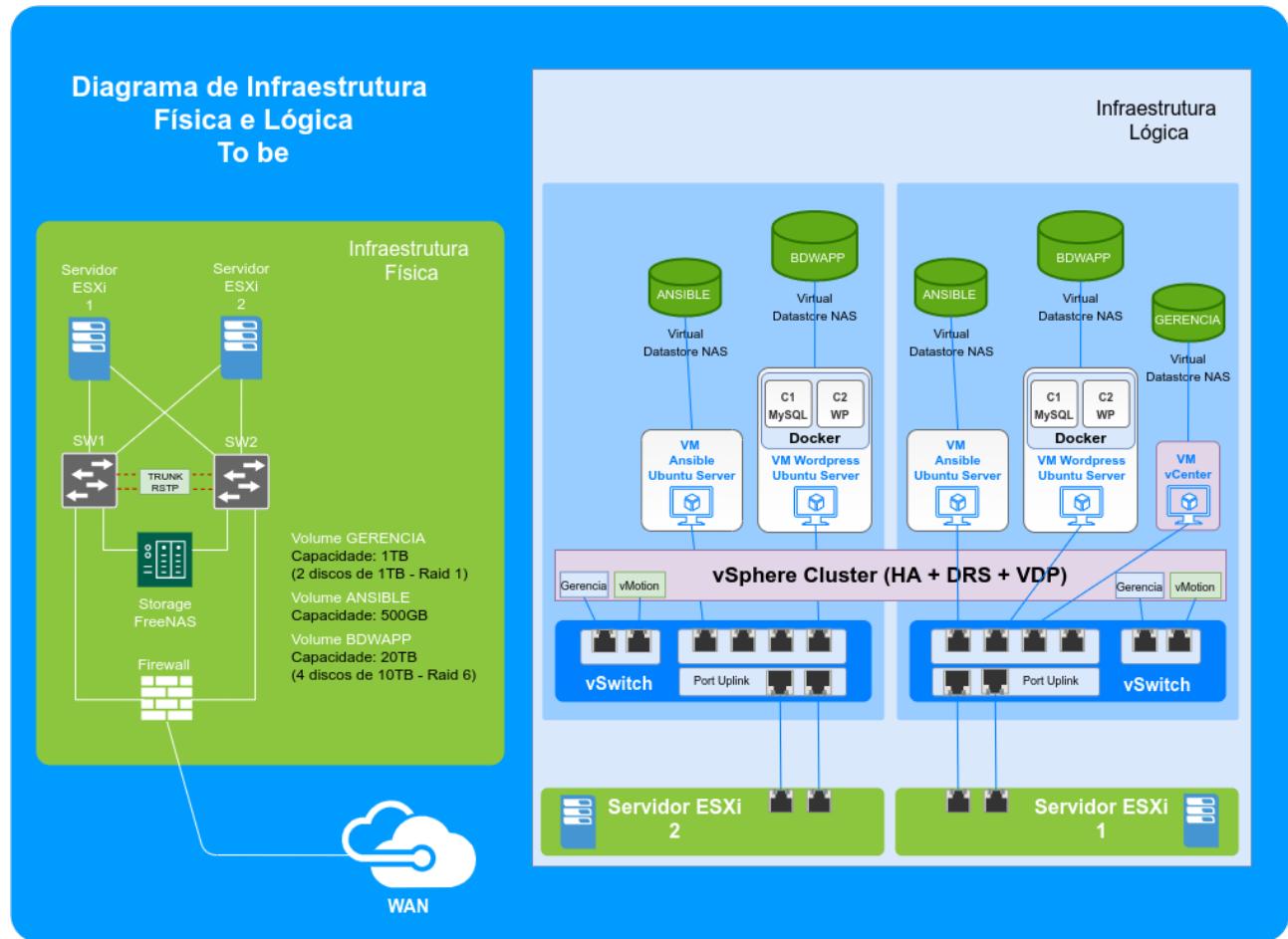
- Rach 19" 44u
 - ✓ Para que seja acomodado dentro do padrão todos os componentes que formam um data center (Rede, Computação e Armazenamento)
- Servidor ESXi
 - 2 Servidores PowerEdge R740 Rack 2U
 - Sistema Operacional: ESXi 6.7
 - Cpu: Intel® Xeon® Gold 5215 de 2,2 GHz, 10 núcleos/20 threads
 - Memória RAM: 128 GB
 - Armazenamento Interno: 480 GB utilizáveis / 960 GB total
 - ESXi: 480 GB de espaço (2 discos de 480GB – Raid 1)
 - ✓ Visa garantir a disponibilidade da instalação do ESXi, mesmo em caso de falha de 1 discos, proporcionando redundância total.
 - 4 Adaptadores de Rede: Gigabit Ethernet
 - ✓ Visa garantir a disponibilidade no uplink
 - Fonte de Alimentação Dupla, Hot-Plug, Redundante (1+1) 750 W
 - ✓ Visa garantir a disponibilidade do servidor em caso de falha da fonte de alimentação.

- Storage NAS
 - Sistema Operacional: FreeNAS 12.0
 - Cpu: Intel® Xeon® 8 núcleos
 - Memória RAM: 64 GB
 - Armazenamento:
 - GERENCIA: 1TB de espaço (2 discos de 1TB – Raid 1) *
 - BDWAPP: 20TB de espaço (4 discos de 10TB – Raid 6) *
 - ANSIBLE: 500GB de espaço (1 discos) *
 - ✓ Devido a natureza da aplicação do cliente, devemos garantir a alta disponibilidade da aplicação. Dessa forma, optamos por escolher o sistema Raid 6 (com 4 discos) que nos permitirá resistir a uma falha de até 2 discos simultâneos sem que haja perda de dados na aplicação e Raid 1 (2 discos espelho) para garantir redundância no gerenciamento dos servidores.

* Dentro do Storage NAS haverá 2 volumes de cada (2x GERENCIA, 2x BDWAPP e 2x ANSIBLE), pois há 2 servidores ESXi.

- Switch
 - 2 Cisco SG350X-24P Stackable Managed Switches
 - ✓ Visa garantir alta disponibilidade em camada 2
 - Sistema Operacional: IOS versão 2.5.8.12
- Firewall Netgate XG-7100 1U
 - Sistema Operacional: PFsense
 - Cpu: Intel "Denverton" Atom C3558, 2,2 GHz, 4 núcleos
 - Memória: 8GB

Diagrama da infraestrutura Física e Lógica - To Be



Detalhes da infraestrutura Física e Lógica - To Be

Como podemos observar no [diagrama](#), os cuidados com a segurança e resiliência da rede começam na infraestrutura física. Um firewall de proteção executará um controle ativo, constante e proativo das ameaças, seguido de 2 switches e 2 servidores ESXi para prover uma rede tolerante a falhas e com alta disponibilidade.

Da infraestrutura física para lógica, temos um cluster vSphere com 2 hosts ESXi. Cada host possuindo 2 conexões de uplink para fornecer um maior grau de disponibilidade. A arquitetura proposta contempla em cada host 2 VMs. Uma dedicada a automação (que será o servidor Ansible) e outra dedicada a aplicação. A proposta da VM de aplicação é fornecer um ambiente em container através do Docker. Sendo dois containers, um para o banco de dados e outro para o wordpress (apache e php). Dessa forma, mantendo os ambientes isolados. O cluster vSphere estará compartilhando os Storages NAS. Onde temos três volumes, cada qual para sua finalidade. Um volume para o servidor Ansible, outro para a aplicação e seus dados (aqui temos uma estrutura de armazenamento no sistema Raid 6 – 4 discos, fornecendo um altíssimo grau de redundância entre os discos para proteger os dados. Podendo perder até 2 discos simultâneos sem que haja prejuízos para a aplicação). E por fim, um volume dedicado a gerencia do cluster (vCenter).

Abrangendo os recursos vSphere, temos algumas soluções (vMotion, HA, DRS, vMotion e VDP) que ajudarão na segurança dos dados e alta disponibilidade. As soluções contemplam: Migração em tempo real de máquinas virtuais entre servidores e switches virtuais, sem interrupção para usuários ou perda de serviço; Balanceamento de carga automatizado usando as características de armazenamento para determinar o melhor local para os dados de uma máquina virtual; Proteção de failover uniforme e econômica contra interrupções de hardware e sistema operacional; Gerenciamento centralizado e eficiente de tarefas de backup.

O resultado é um ambiente de TI simplificado, sempre disponível, mais econômico e resiliente, automatizado e que atende melhor às necessidades dinâmica do negócio.

Infraestrutura - As Is

Com o objetivo de validar as funcionalidades dos softwares definidos para compor a solução de SDDC, bem como o nascimento e funcionamento da aplicação. Implementamos este projeto em um ambiente simulado de testes, com requisitos mínimos de hardware, sem nenhum tipo de custo, apenas com o intuito de validar os requisitos funcionais (e não de performance) do SDDC.

Utilizamos os seguintes requisitos de hardware para simular infraestrutura física:

Hardware Físico Real sobre o qual virtualizamos o ambiente:

- Sistema operacional: Fedora 35
- Cpu: Intel core i7 com 8 núcleos
- Memória RAM: 20 GB
- Armazenamento: 1 TB SSD
- Adaptador de Rede: Gigabit Ethernet

Hardware Físico Simulado Virtualizador tipo 2:

- Vmware Workstation 16 Pro

Virtualizador tipo 1: instalado no hardware físico simulado:

- Sistema operacional: ESXi 6.7
- vCpu: 4 núcleos
- Memória: 14 GB
- Armazenamento: 100 GB
- Adaptador de Rede: 1 vnic (nat)

VM Storage NAS: Instalado dentro do Vmware Workstation

- Sistema operacional: FreeNAS 9.1.2.8
- vCpu: 1 núcleo
- Memória RAM: 2 GB
- Armazenamento: 80 GB utilizáveis / 120 GB total
 - Volume BDWAPP: 40GB de espaço (4 discos de 20GB – Raid 6)
 - Volume ANSIBLE: 40GB de espaço (1 discos)
- Adaptador de Rede: 1 vnic (nat)

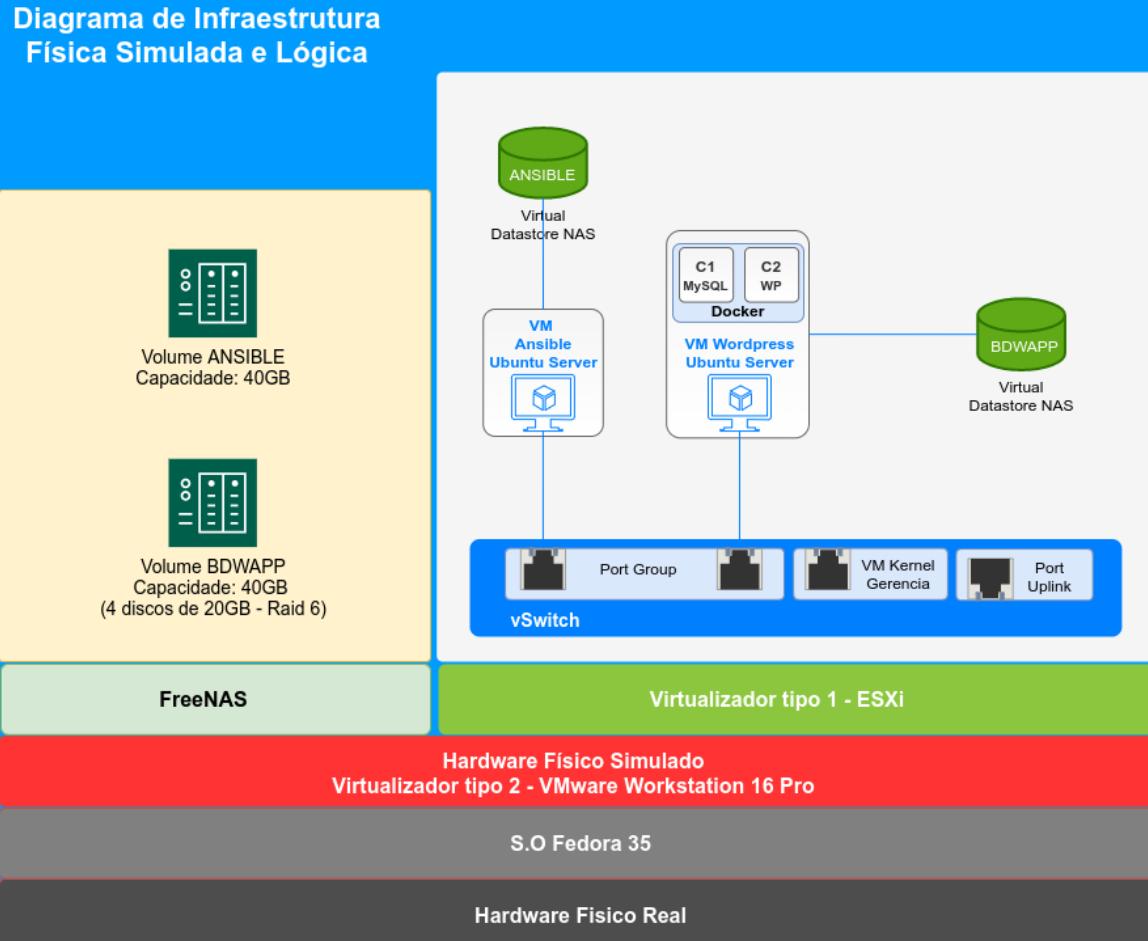
VM BDWAPP: Instalado dentro do ESXi

- Sistema operacional: Linux Ubuntu Server 20.04
- vCpu: 2 núcleo
- Memória RAM: 4 GB
- Armazenamento: Virtual datastore NAS - BDWAPP
- Adaptador de Rede: 1 vnic (nat)

VM ANSIBLE: Instalado dentro do ESXi

- Sistema operacional: Linux Ubuntu Server 20.04
- vCpu: 2 núcleo
- Memória RAM: 4 GB
- Armazenamento: Virtual datastore NAS - ANSIBLE
- Adaptador de Rede: 1 vnic (nat)

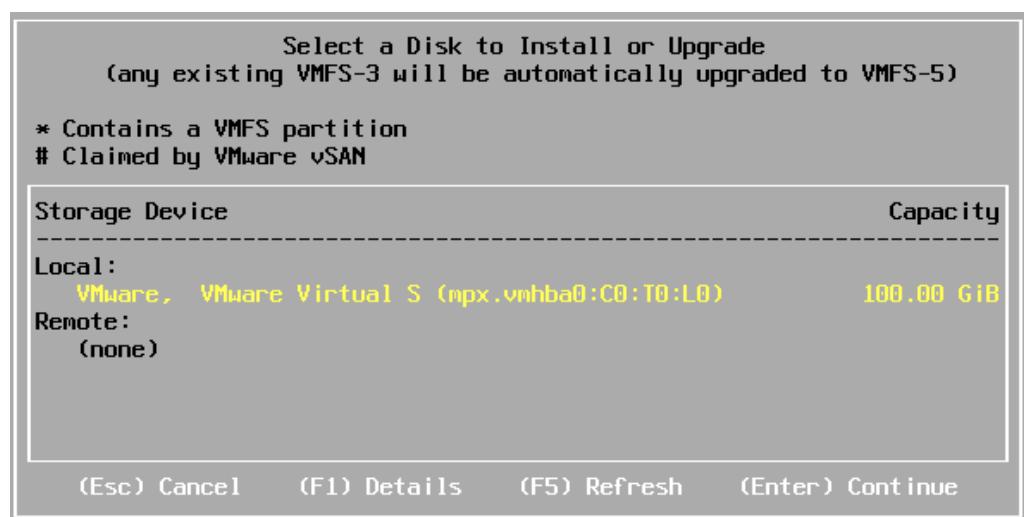
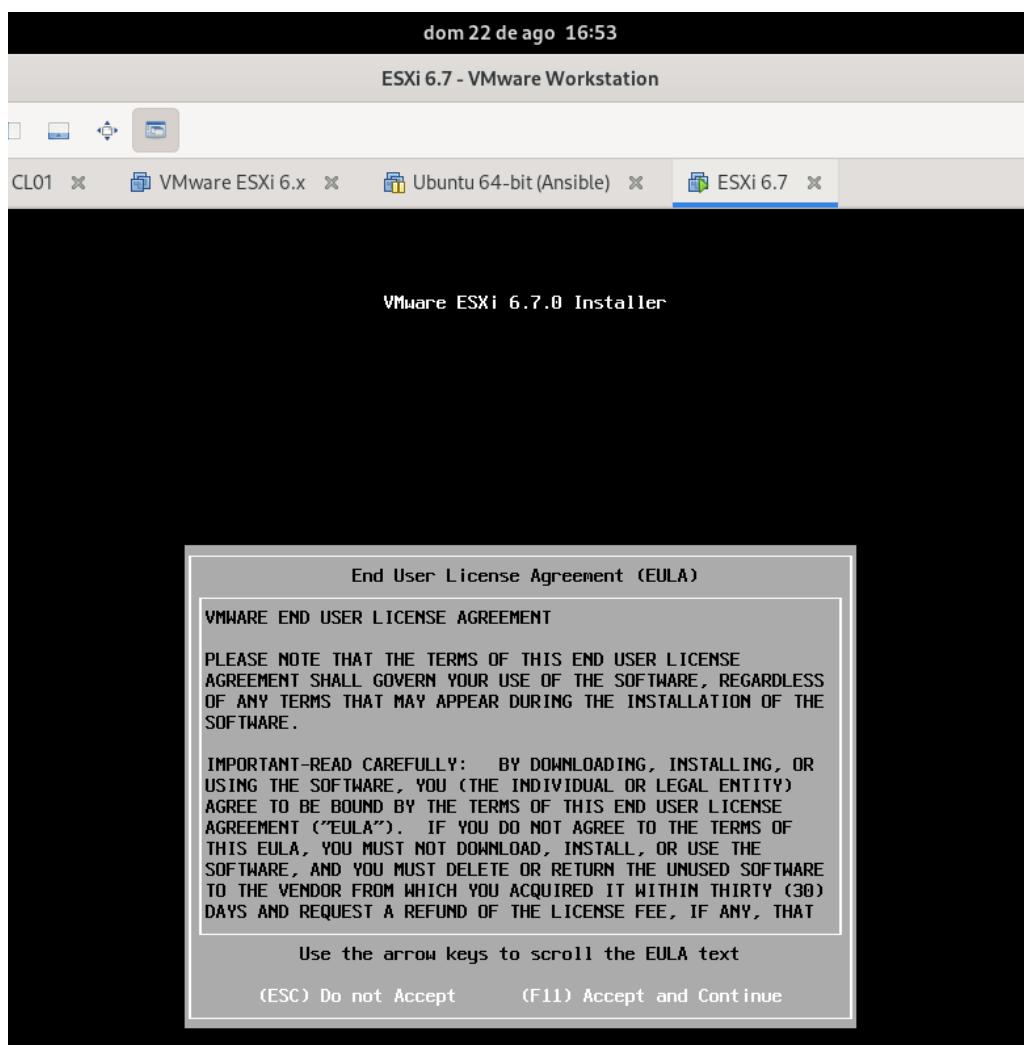
Diagrama da infraestrutura Física Simulada - As Is



Nas páginas seguintes estaremos apresentando os processos de instalação dos elementos que compõe o SDDC que foi montado neste ambiente simulado, bem como a infraestrutura para receber a aplicação.

Processo de instalação do host VMware ESXi

O processo de instalação é muito simples. Basicamente aceitar os termos, selecionar o disco onde será feito a instalação, o layout do teclado, a senha de root e iniciar a instalação.



Please select a keyboard layout

Belgian

Brazilian

Croatian

Czechoslovakian

Danish

Estonian

Finnish

Use the arrow keys to scroll.

(Esc) Cancel (F9) Back (Enter) Continue

Enter a root password

Root password: ****

Confirm password: ****_

Passwords match.

(Esc) Cancel (F9) Back (Enter) Continue

Confirm Install

The installer is configured to **install** ESXi 6.7.0 on:
mpx.vmhba0:C0:T0:L0.

Warning: This disk will be repartitioned.

(Esc) Cancel (F9) Back (F11) Install

Após a instalação concluída é nos apresentado a tela inicial do ESXi. Nela podemos customizar as configurações, visualizar logs, desligar e reiniciar o virtualizador.



Como pode ser visto na imagem acima, a configuração padrão vem com o IPv4 em DHCP. Como em um ambiente real isso é uma prática inviável, antes de mais nada deveremos reconfigurar o IP de dinâmico para estático.

Para configurar pressionamos a tecla “F2”. Após digitarmos a senha, nos será levado a uma tela de configuração onde selecionaremos a opção “IPv4 Configuration”. Aqui setaremos o endereçamento IPv4 para estático.

IPv4 Configuration

This host can obtain network settings automatically if your network includes a DHCP server. If it does not, the following settings must be specified:

- () Disable IPv4 configuration for management network
- () Use dynamic IPv4 address and network configuration
- (o) Set static IPv4 address and network configuration:

IPv4 Address	[192.168.1.11]
Subnet Mask	[255.255.255.0]
Default Gateway	[192.168.1.254]

<Up/Down> Select <Space> Mark Selected

<Enter> OK <Esc> Cancel

Após fixarmos o IP é só confirmar com a tecla “enter” e aplicar as configurações com “Y”.
Após essa alteração na configuração, o IP já nos é apresentado como (STATIC).

VMware ESXi 6.7.0 (VMKernel Release Build 15160138)
VMware, Inc. VMware7.1
4 x Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz
12 GiB Memory

To manage this host go to:
[http://192.168.1.11/ \(STATIC\)](http://192.168.1.11/)
[http://\[fe80::20c:29ff:fe58:f706\]/ \(STATIC\)](http://[fe80::20c:29ff:fe58:f706]/)
[http://\[2004:d55:439b:8200:20c:29ff:fe58:f706\]/ \(AUTOCONF\)](http://[2004:d55:439b:8200:20c:29ff:fe58:f706]/)

<F2> Customize System/View Logs

<F12> Shut Down/Restart

Agora de posse do endereço IP, basta digitarmos ele na barra de endereços de um navegador web e acessar a página do ESXi.

The screenshot shows the VMware ESXi host interface. At the top, there's a navigation bar with links for 'Get vCenter Server', 'Create/Register VM', 'Shut down', 'Reboot', 'Refresh', and 'Actions'. The main title is 'localhost.localdomain' with the sub-title 'Version: 6.7.0 Update 3 (Build 15160138) State: Normal (not connected to any vCenter Server) Uptime: 0 days'. On the right, resource usage bars show CPU (1% used), Memory (11% used), and Storage (28% used). Below this, the 'Hardware' section lists the host's configuration:

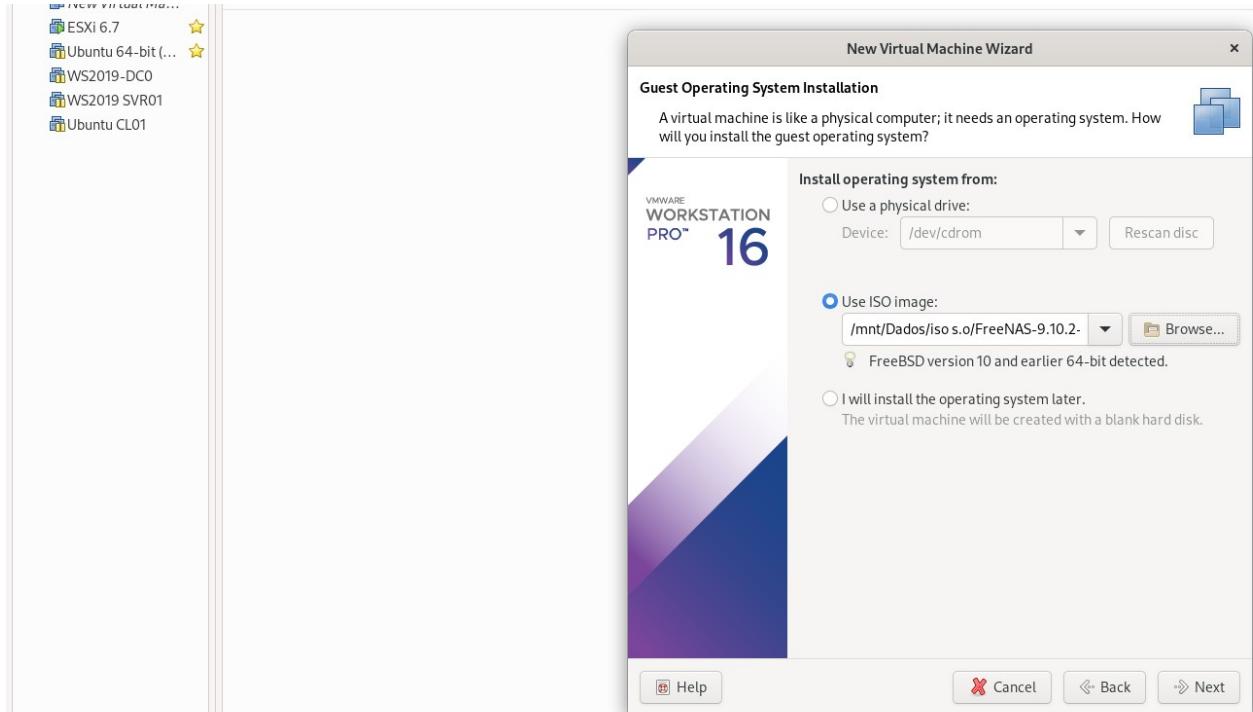
Manufacturer	VMware, Inc.
Model	VMware7,1
CPU	8 CPUs x Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz
Memory	12 GB
Persistent Memory	0 B
Virtual flash	0 B used, 0 B capacity
Networking	Hostname: localhost.localdomain
IP addresses	1. vmk0: 192.168.1.11 2. vmk0: fe80::20c:29ff:fe58:f706 3. vmk0: 2804:d55:439b:8200:20c:29ff:fe58:f706
DNS servers	1. 192.168.1.254
Default gateway	192.168.1.254
IPv6 enabled	Yes
Host adapters	1
Networks	Name: VM Network VMs: 1

The 'Storage' section shows three physical adapters and one datastore:

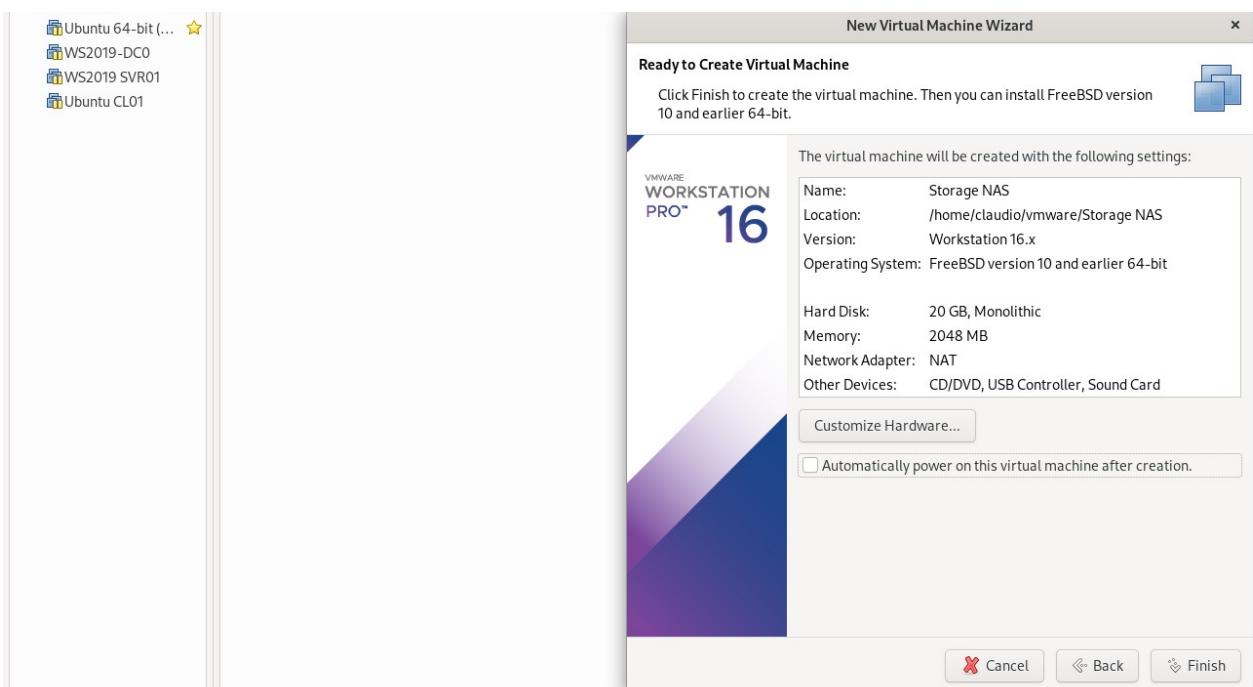
Physical adapters	3
Datastores	Name: datastore1 Type: VMFS6 Capacity: 72.5 GB Free: 52.23 GB

Processo de instalação do FreeNAS

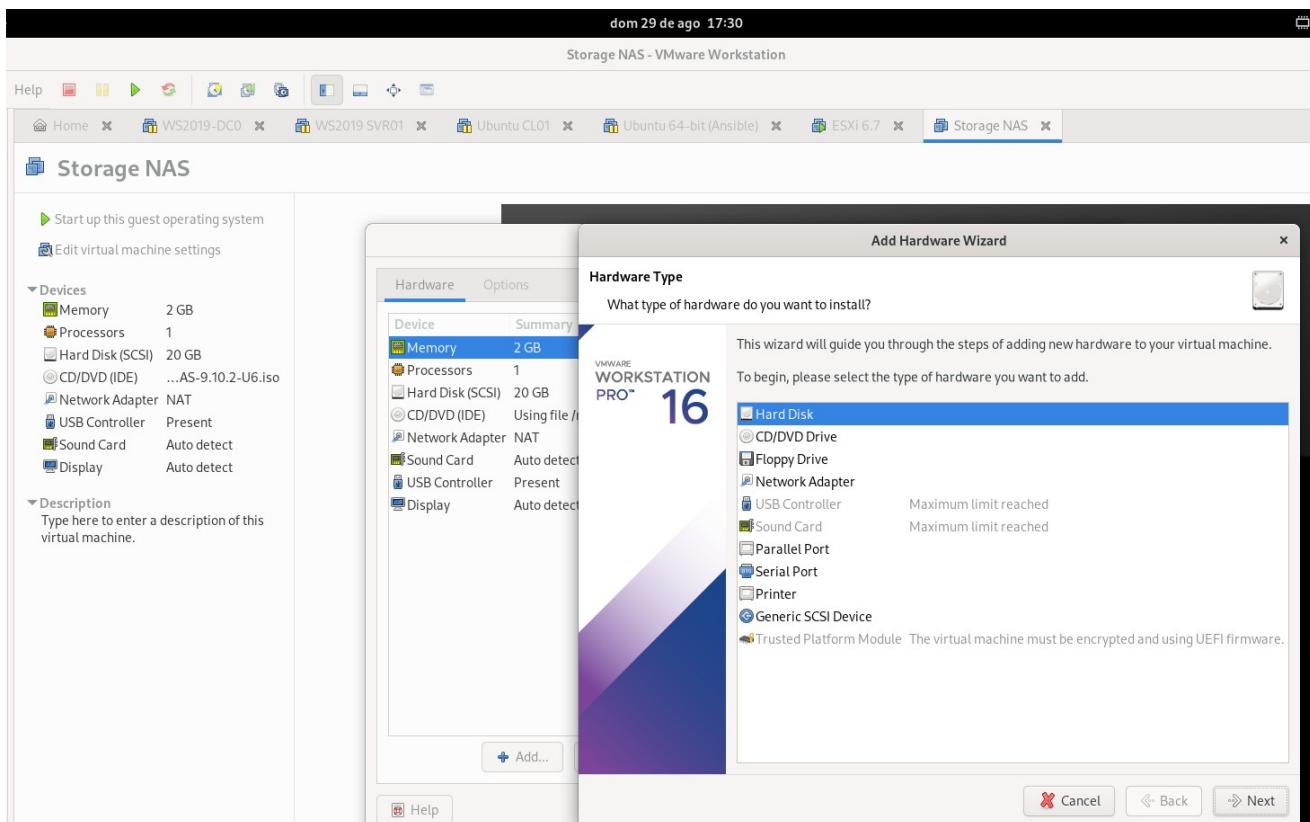
O início da instalação segue o padrão normal de qualquer instalação no vmware. Seleciona a ISO, escolhe o nome, armazenamento e as configurações de hardware.



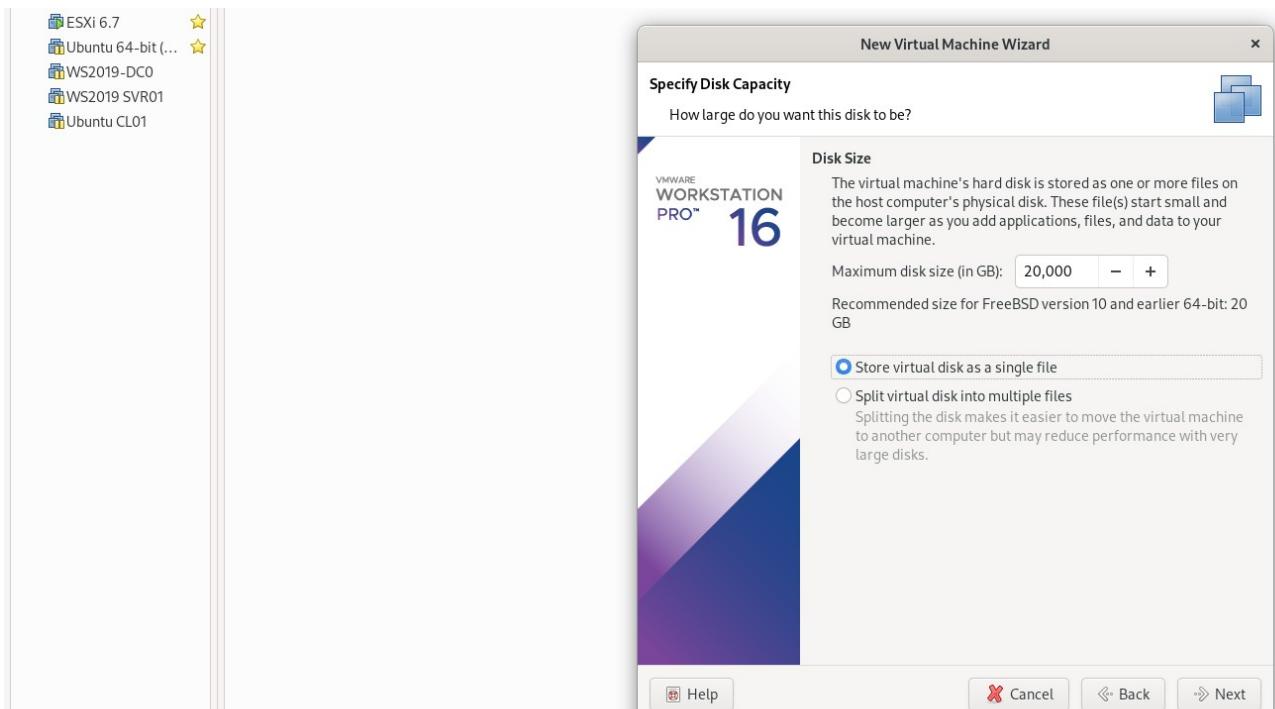
Após configurado, desmarcamos a opção para ligar a VM automaticamente, pois antes iremos adicionar mais 5 hds para criação de 2 volumes.



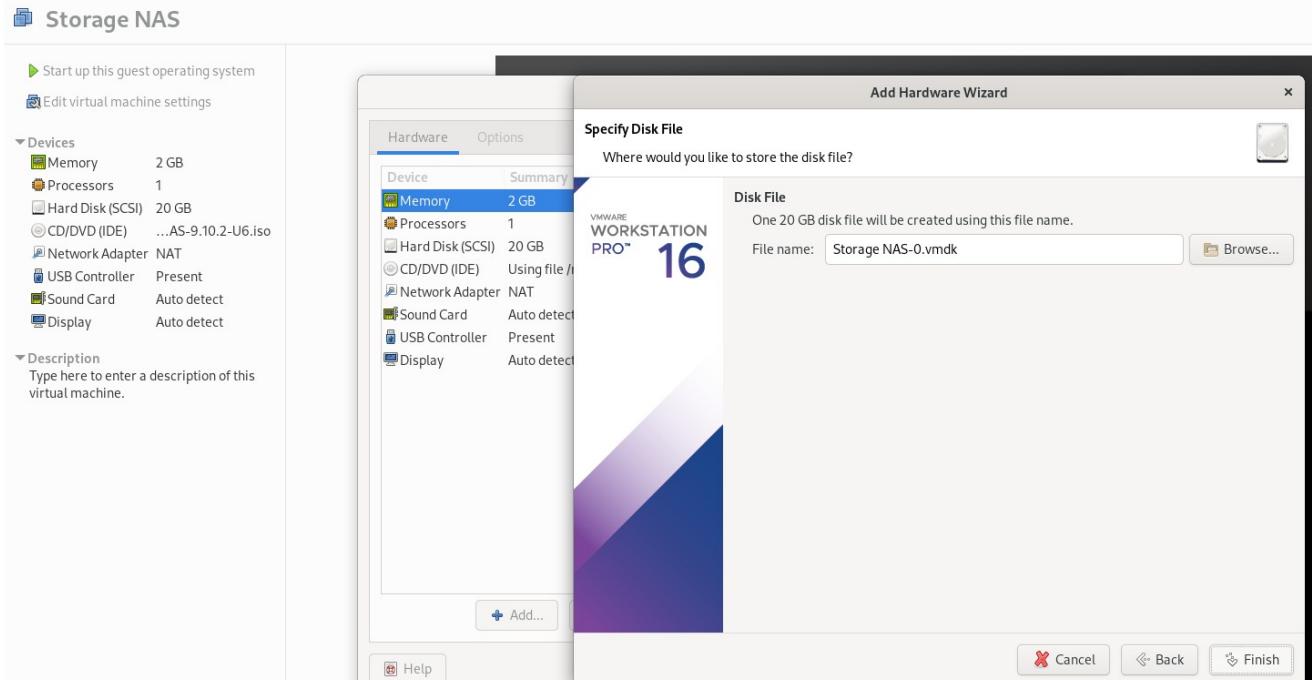
Para isso, vamos clicar em “**Edit virtual machine settings**” e escolher a opção “**Hard Disk**” e dar um “**Next**”.



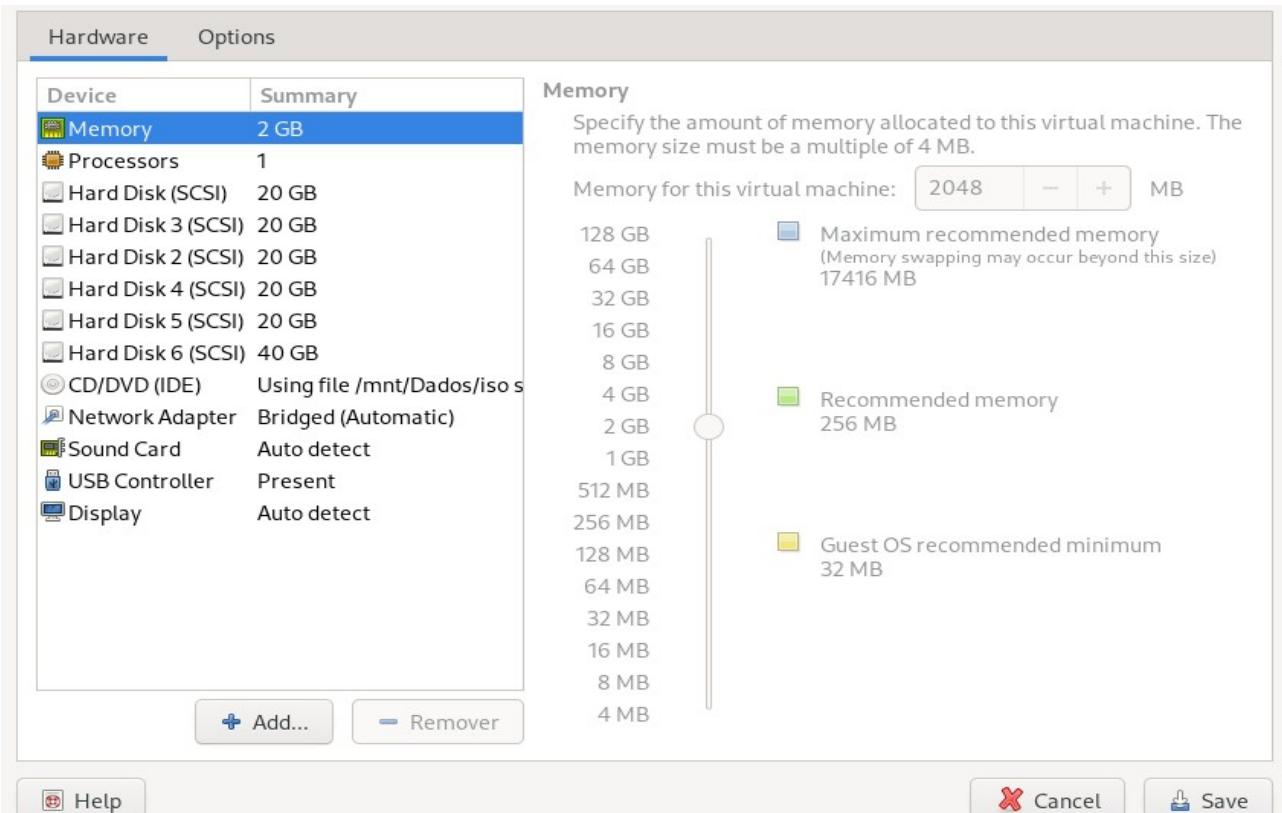
Seguimos com as configurações padrões do instalador (SCSI e Criar um novo disco virtual) até chegar na configuração do tamanho do disco.



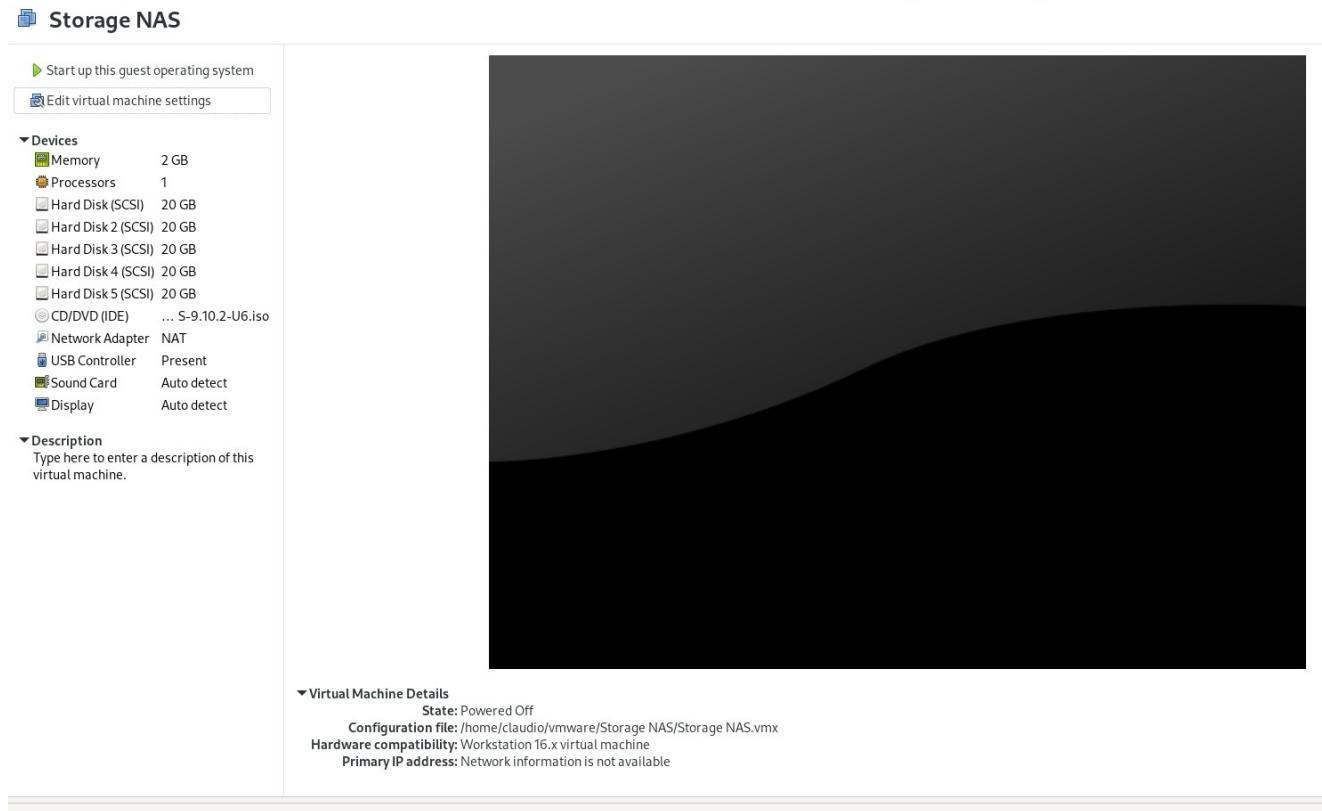
Após escolher o tamanho do disco, a adição do mesmo já está concluído.



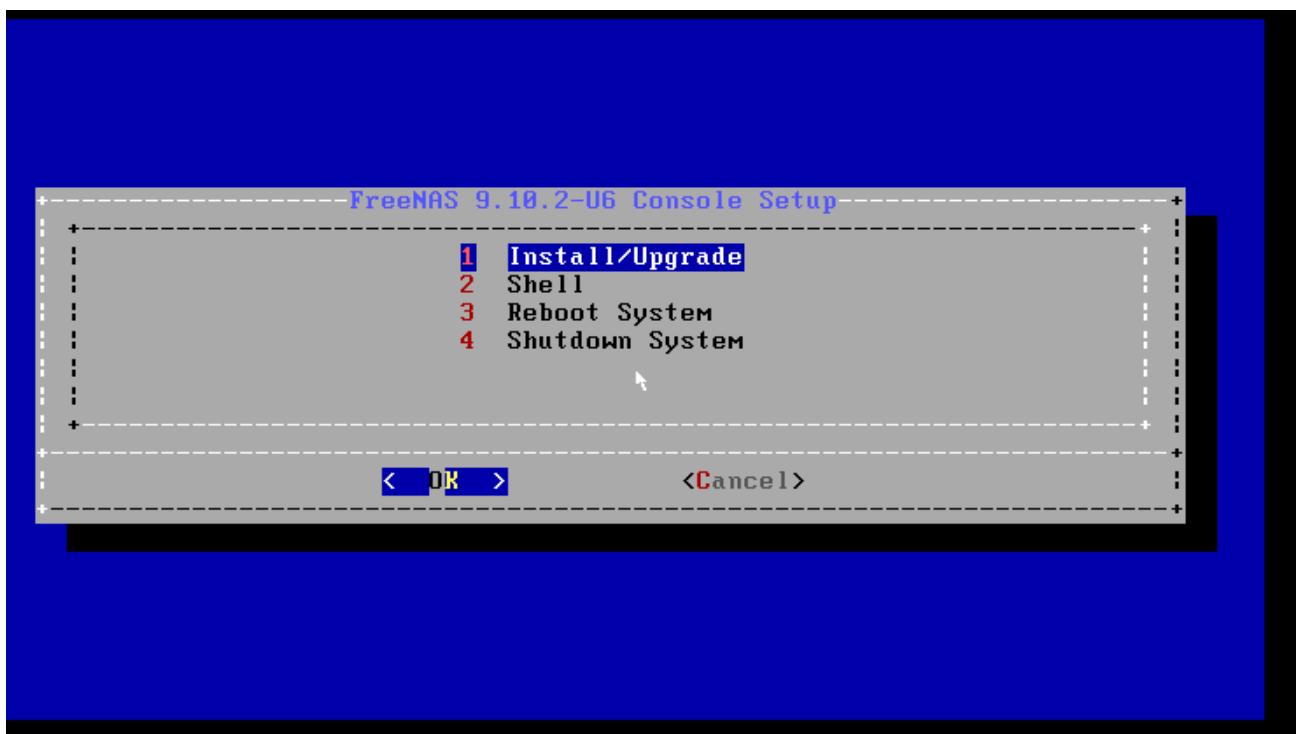
Agora repetiremos o processo por mais 4 vezes até obtermos os 5 discos (4 de 20GB e 1 de 40GB). Sendo 4 para formar um volume Raid 6 e 1 para formar um volume único. Feito isso, é só clicar em “**Salvar**”



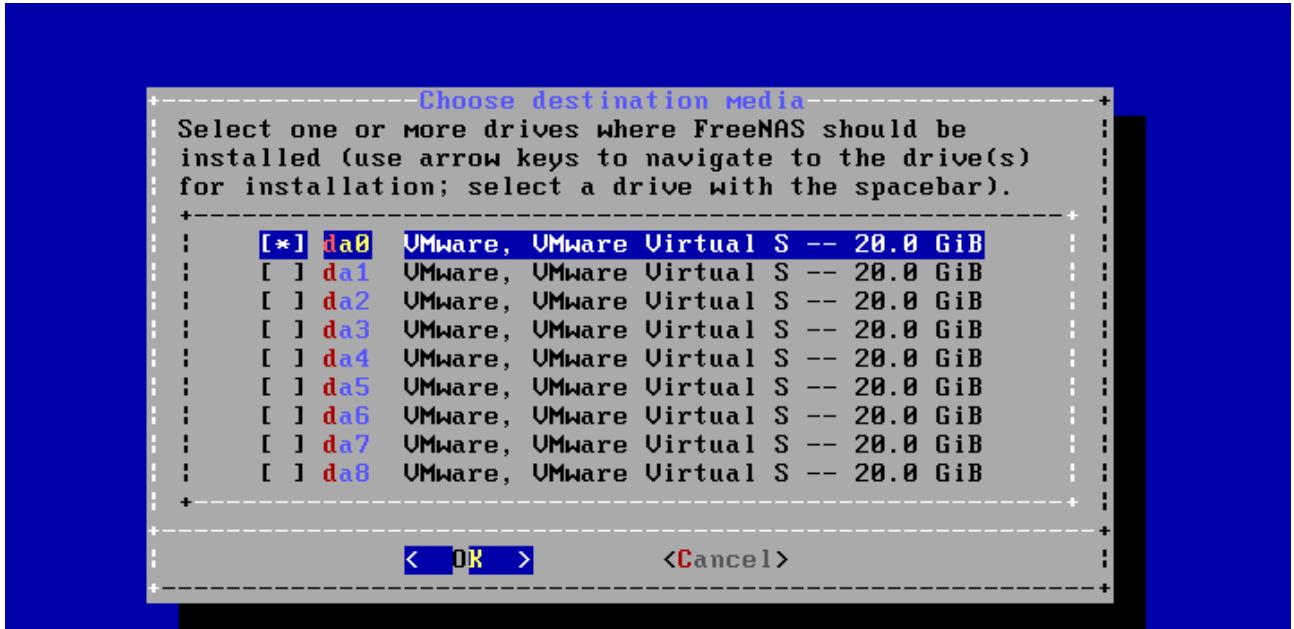
Pronto! Agora vamos dar um “Start” na VM para iniciar a instalação de fato do FreeNAS.



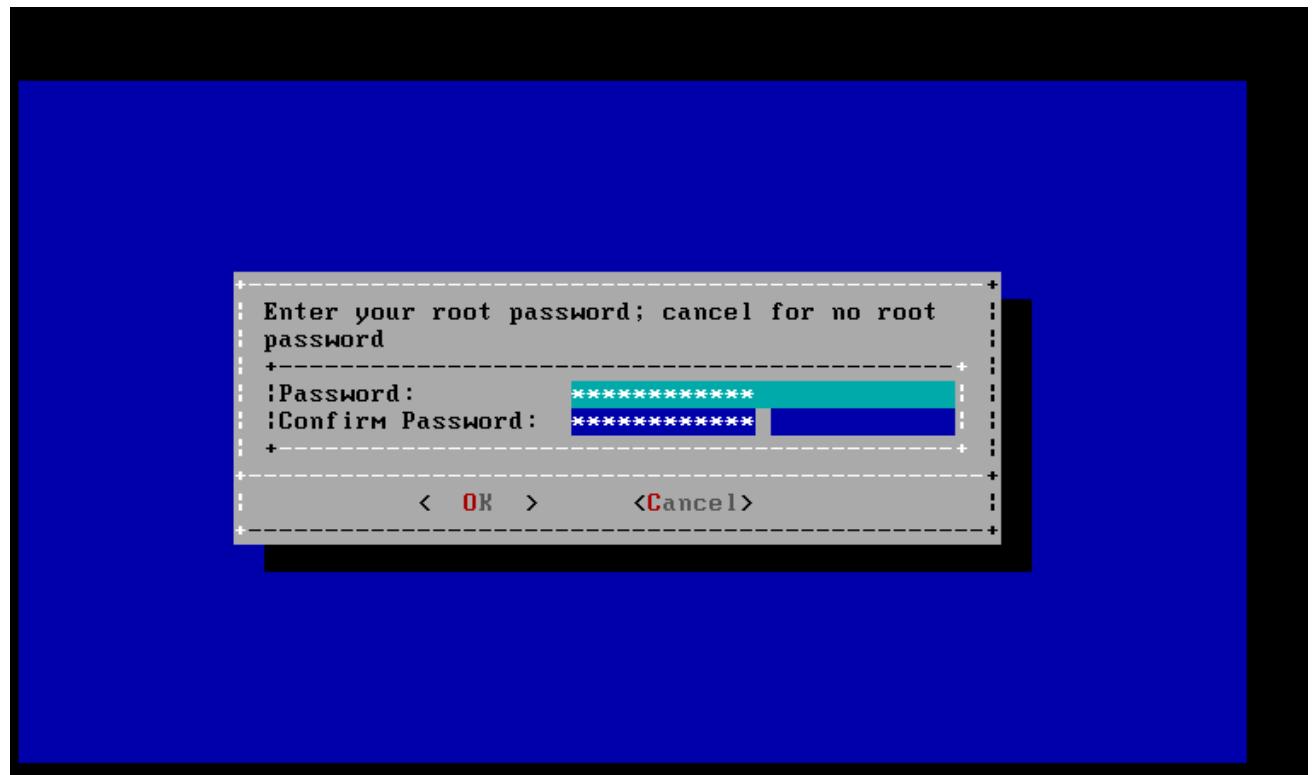
Selecione a opção 1 “install/upgrade” e de um “ok” para iniciar a instalação



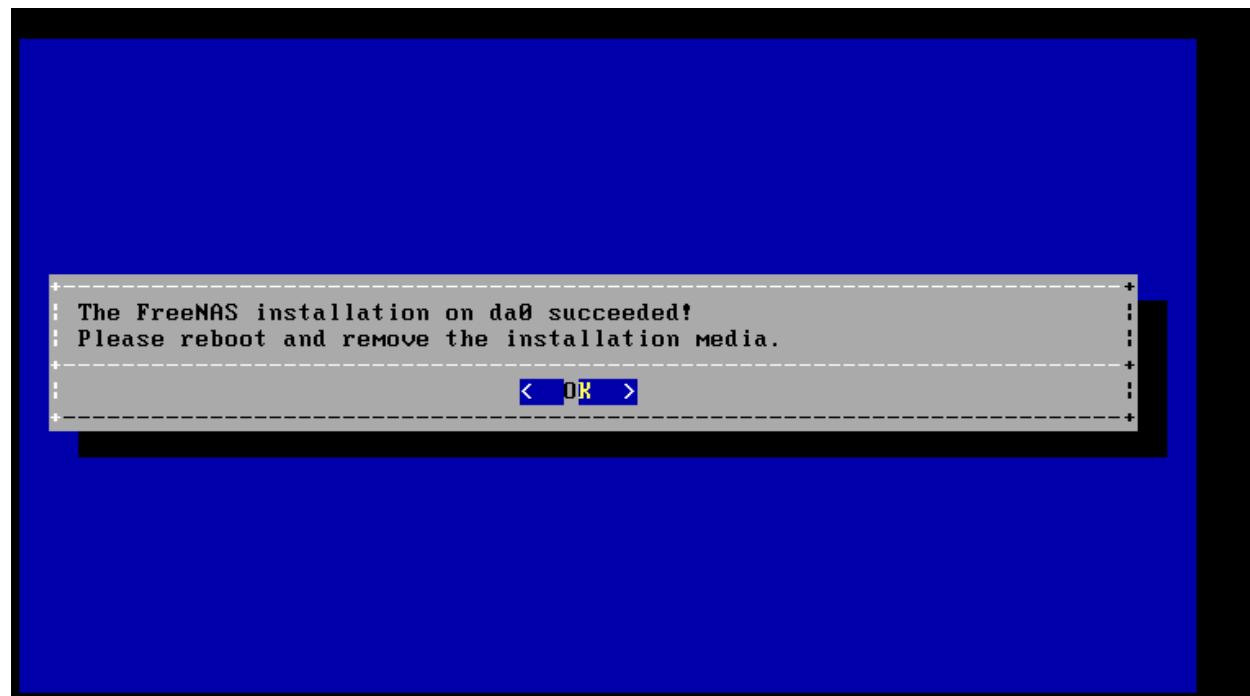
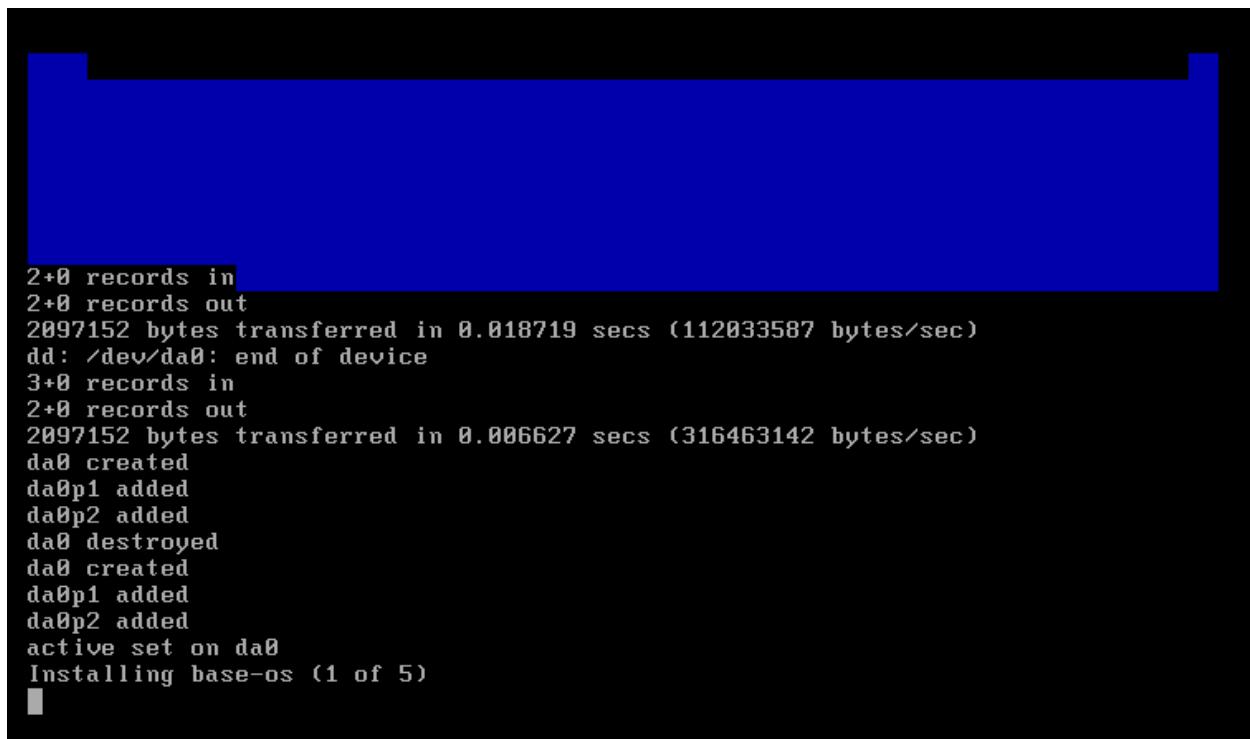
Agora selecione o disco que será instalado o FreeNAS e de um “ok”.



Após confirmar que as partições serão apagadas, seguimos com a instalação e escolheremos a senha de root.



Na próxima tela escolheremos a configuração sugerida para o boot (boot via bios) e a instalação seguirá até sua conclusão.



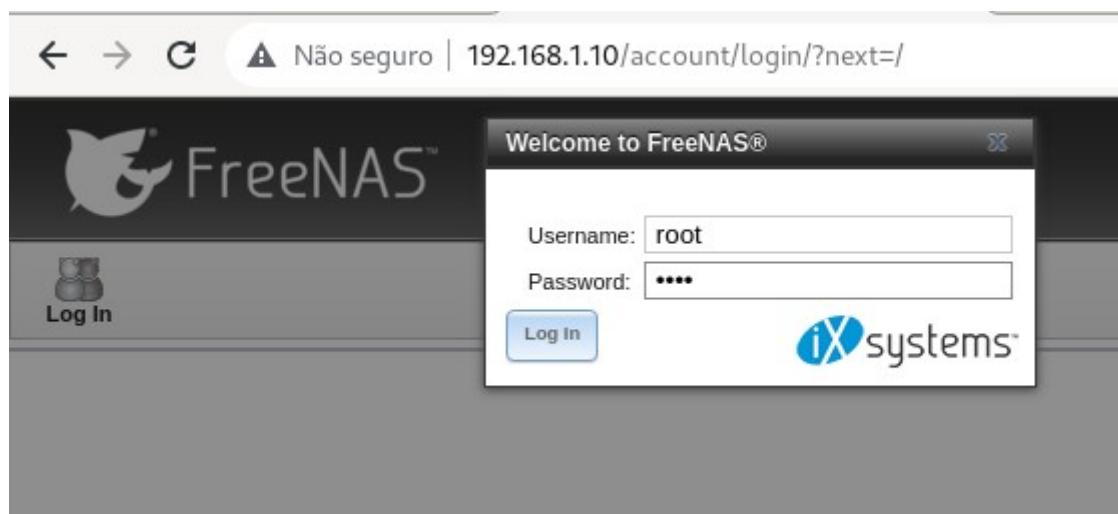
Após a conclusão é só dar um “**reboot**” na maquina e aguardar ela iniciar. Após o reinicio teremos a tela a seguir.

```
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 to factory defaults
9) Shell
10) System Update (requires networking)
11) Create volume backup
12) Restore volume from a backup
13) Reboot
14) Shutdown

You may try the following URLs to access the web user interface:
http://192.168.1.10

Enter an option from 1-14: ■
```

Agora abriremos o navegador web e digitaremos o IP fornecido na etapa anterior. Colocaremos a senha que cadastramos na instalação e já estaremos prontos para iniciar a configuração dos volumes do storage.



Processo de Configuração dos Volumes no FreeNAS

Tela inicial do FreeNAS

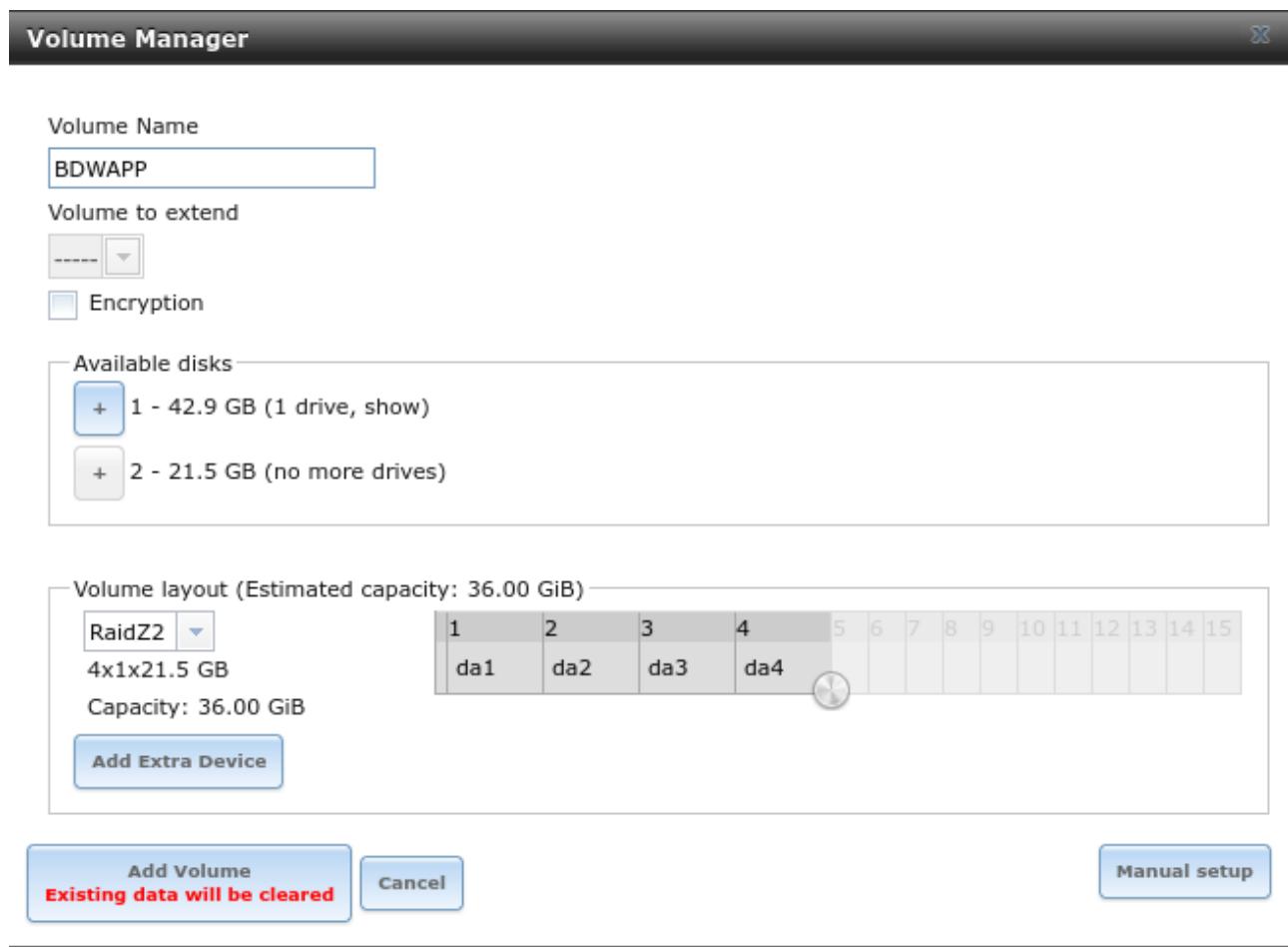
The screenshot shows the FreeNAS web interface. At the top, there's a navigation bar with icons for Account, System, Tasks, Network, Storage, Directory, Sharing, Services, Plugins, Jails, Reporting, Wizard, and OK. The URL in the address bar is "192.168.1.10". The main content area has a sidebar on the left with "expand all" and "collapse all" buttons, listing various system components like Account, System, Tasks, Network, Storage, etc. The main panel is titled "System Information" and displays the following details:

Hostname	freenas.local	Edit
Build	FreeNAS-9.10.2-U6 (561f0d7a1)	
Platform	Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz	
Memory	2018MB	
System Time	Wed Nov 24 17:25:02 PST 2021	
Uptime	5:25PM up 11 mins, 0 users	
Load Average	0.10, 0.24, 0.23	

Vamos iniciar criando um volume para o storage. Para isso, clicaremos em “Storage” e depois em “Volume Manager”.

The screenshot shows the FreeNAS web interface with the "Storage" tab selected in the top navigation bar. The sidebar on the left still lists various system components. The main panel is titled "Storage" and then "Volumes". It includes buttons for "Periodic Snapshot Tasks", "Replication Tasks", "Scrubs", "Snapshots", and "VMware-Snapshot". Below these buttons, there are four smaller buttons: "Volume Manager", "Import Disk", "Import Volume", and "View Disks". A table titled "Name" is shown, with columns for "Used", "Available", "Compression", "Compression Ratio", "Status", "Readonly", and "Comments". A message at the bottom of the table says "No entry has been found".

Nesta tela daremos um nome para o volume e configuraremos a quantidade de discos para fazer o Raid. No nosso caso, como foi adicionado 4 discos, faremos o Raid 6 (aqui usa-se a nomenclatura Raid Z2). Após, é só clicar em “**Add Volume**”.

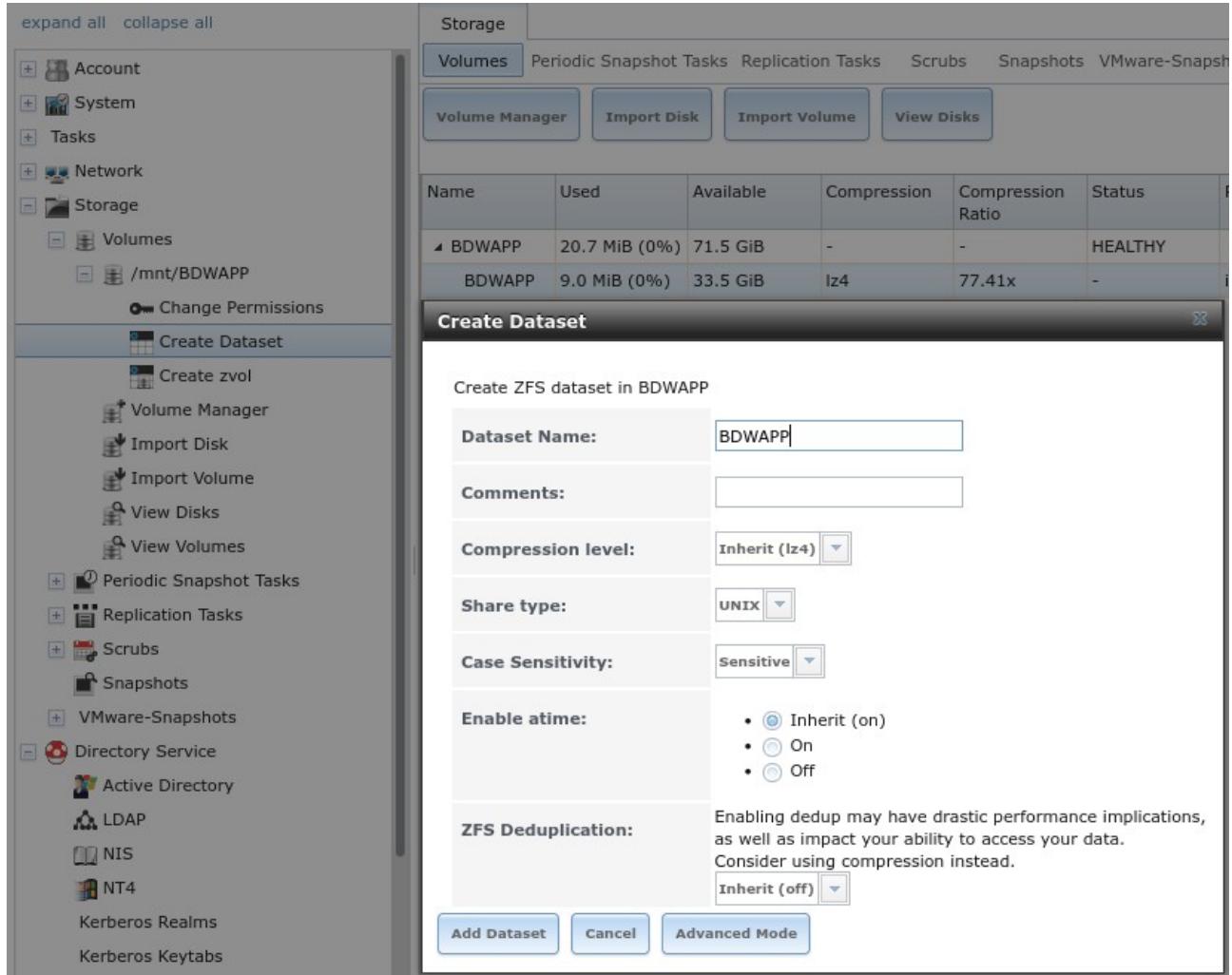


The screenshot shows the 'Storage' interface. The 'Volumes' tab is selected, displaying a list of volumes. The 'BDWAPP' volume is listed with the following details:

Name	Used	Available	Compression	Compression Ratio	Status	Readonly	Comments
BDWAPP	20.7 MiB (0%)	71.5 GiB	-	-	HEALTHY		
BDWAPP	9.0 MiB (0%)	33.5 GiB	lz4	77.41x	-	inherit (off)	

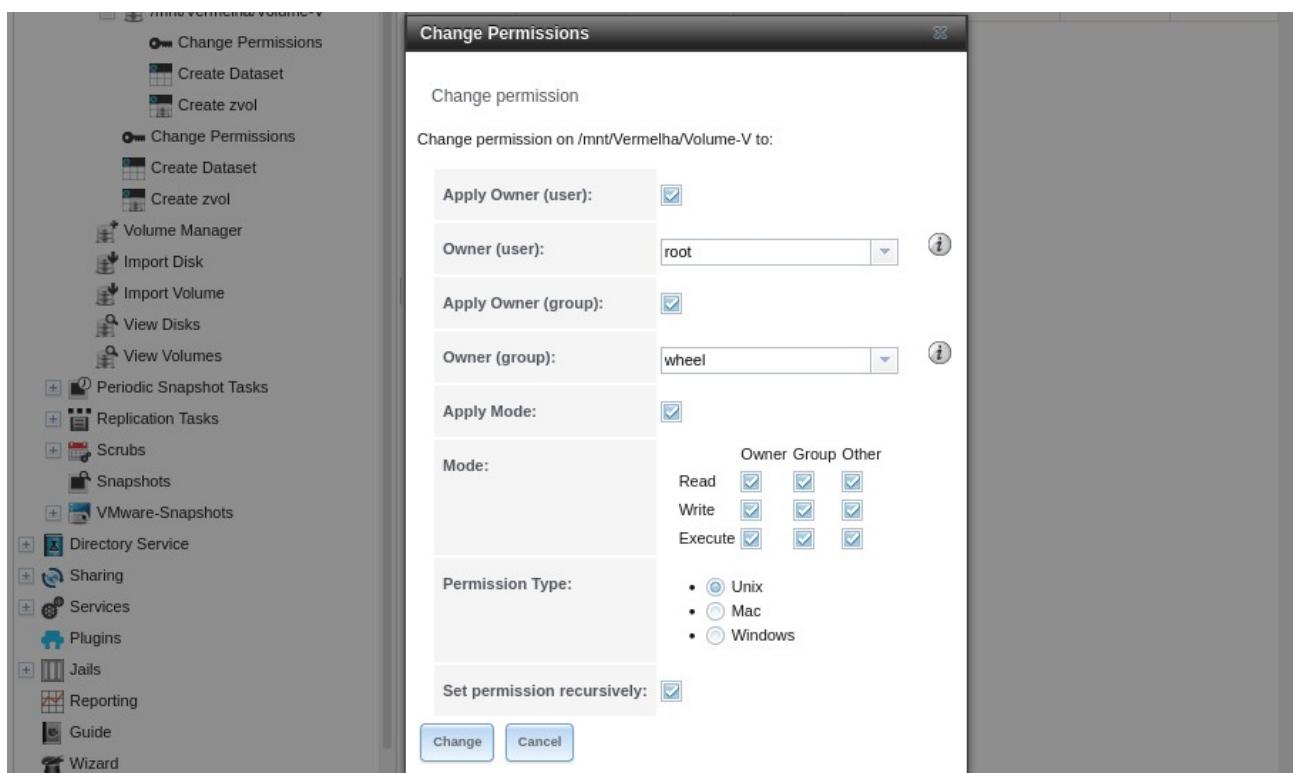
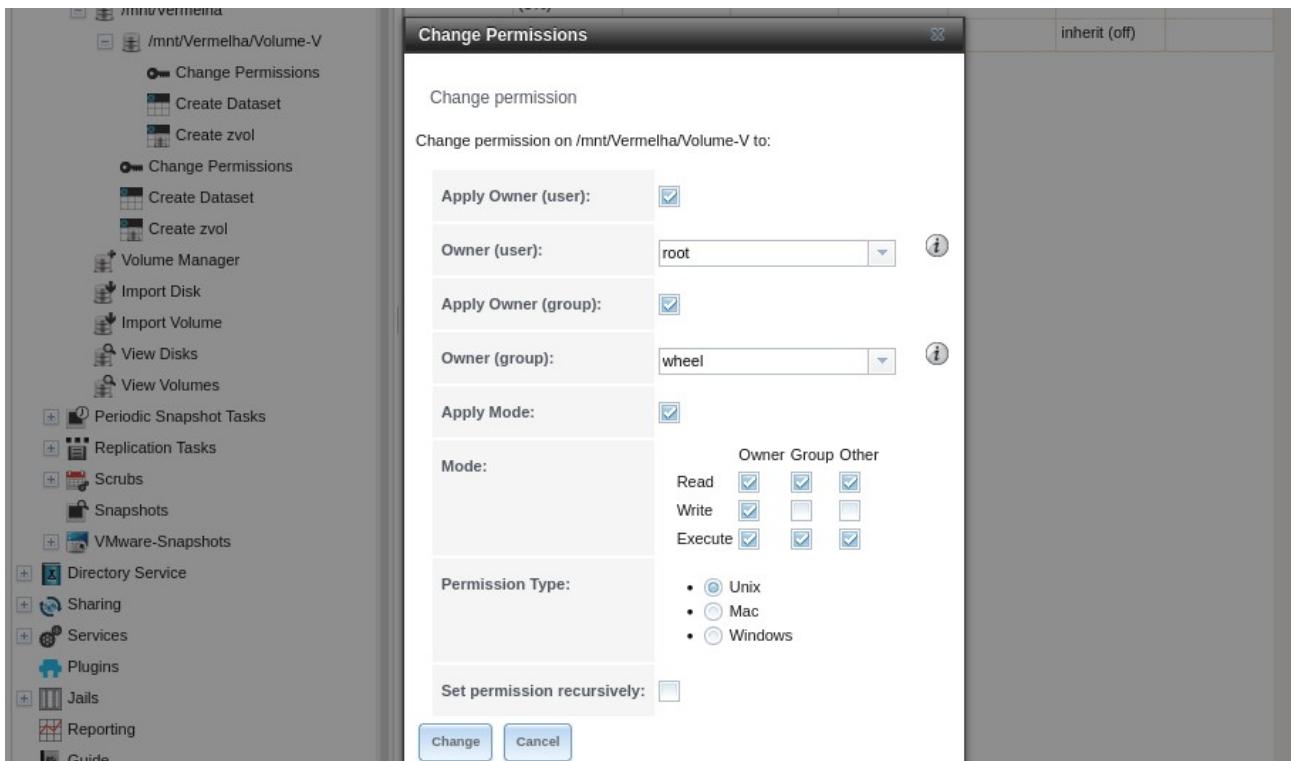
Após a criação do volume, precisamos deixar esse volume visível ao sistema operacional. Para isso criaremos um “**Dataset**” conforme a imagem abaixo.

Para criar um Dataset, clique em “Create Dataset”



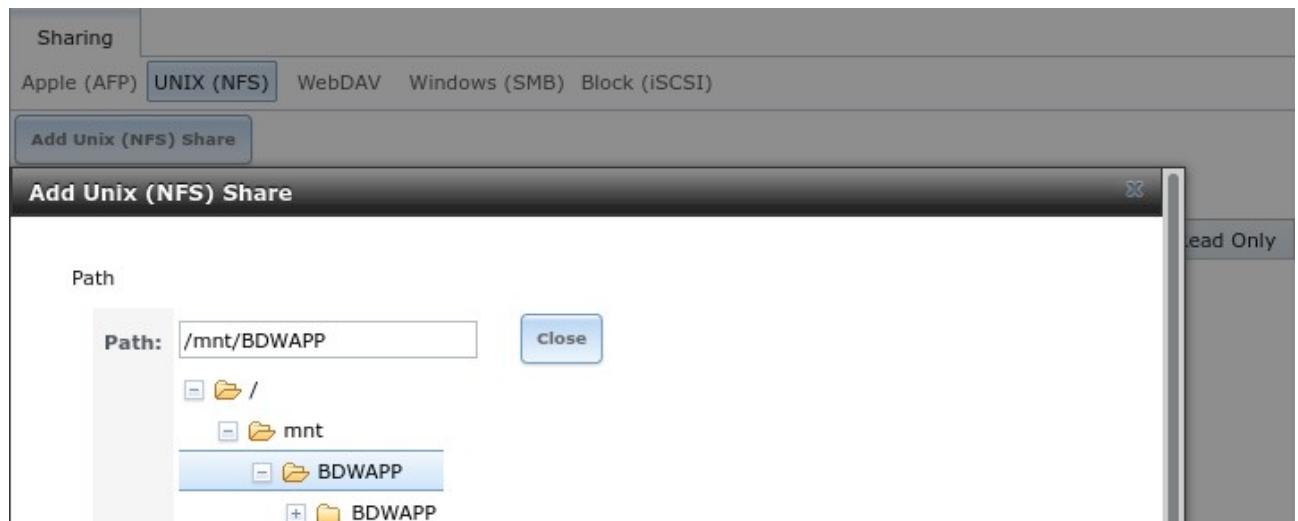
Concluído a criação do Dataset, precisamos configurar as permissões para que o ESXi possa gravar em disco.

Como podemos ver na imagem a seguir, por padrão o freeNAS deixa desabilitado a permissão de escrita. Vamos ativá-las! Para isso vamos até a opção “**change permissions**” e na tela que se abrirá basta clicar nas caixas que não estão marcadas para escrita (white) e depois clicar em “**Change**”.



Agora vamos adicionar o compartilhamento do storage, para que o volume seja encontrado pelo ESXi. Para isso clicamos em “Sharing”, depois em “AddUnix (NFS) Share”.

Uma tela se abrirá e nela selecionaremos o caminho do compartilhamento. Feito isso, basta dar um “ok” e depois um “yes” para habilitar o serviço de **NFS**.



Terminado todas as etapas de criação, configuração e compartilhamento do volume “**BDWAPP**”, vamos executar agora os mesmos procedimentos para o volume “**ANSIBLE**”.

Volume criado, configurado e compartilhado.

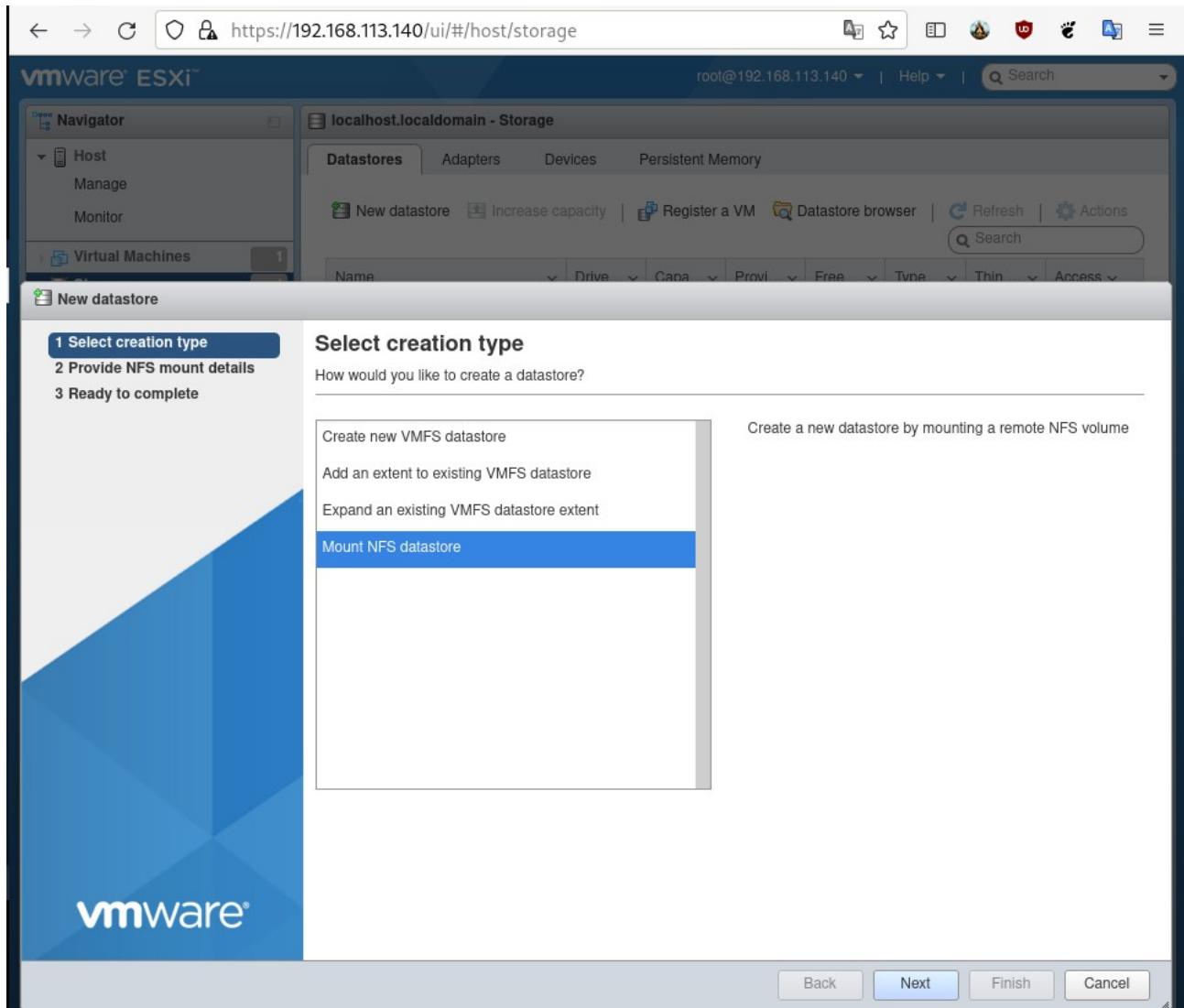
Name	Used	Available	Compression	Compression Ratio	Status	Readonly	Comments
▲ BDWAPP	22.1 MiB (0%)	71.5 GiB	-	-	HEALTHY		
▲ BDWAPP	10.5 MiB (0%)	33.5 GiB	lz4	61.96x	-	inherit (off)	
BDWAPP	127.9 KiB (0%)	33.5 GiB	inherit (lz4)	1.00x	-	inherit (off)	
▲ ANSIBLE	464.0 KiB (0%)	37.7 GiB	-	-	HEALTHY		
▲ ANSIBLE	416.0 KiB (0%)	36.6 GiB	lz4	1.00x	-	inherit (off)	
ANSIBLE	88.0 KiB (0%)	36.6 GiB	inherit (lz4)	1.00x	-	inherit (off)	

Agora que foi terminado o compartilhamento dos volumes via NFS no FreeNAS, é hora de adicionar os datastores no ESXi.

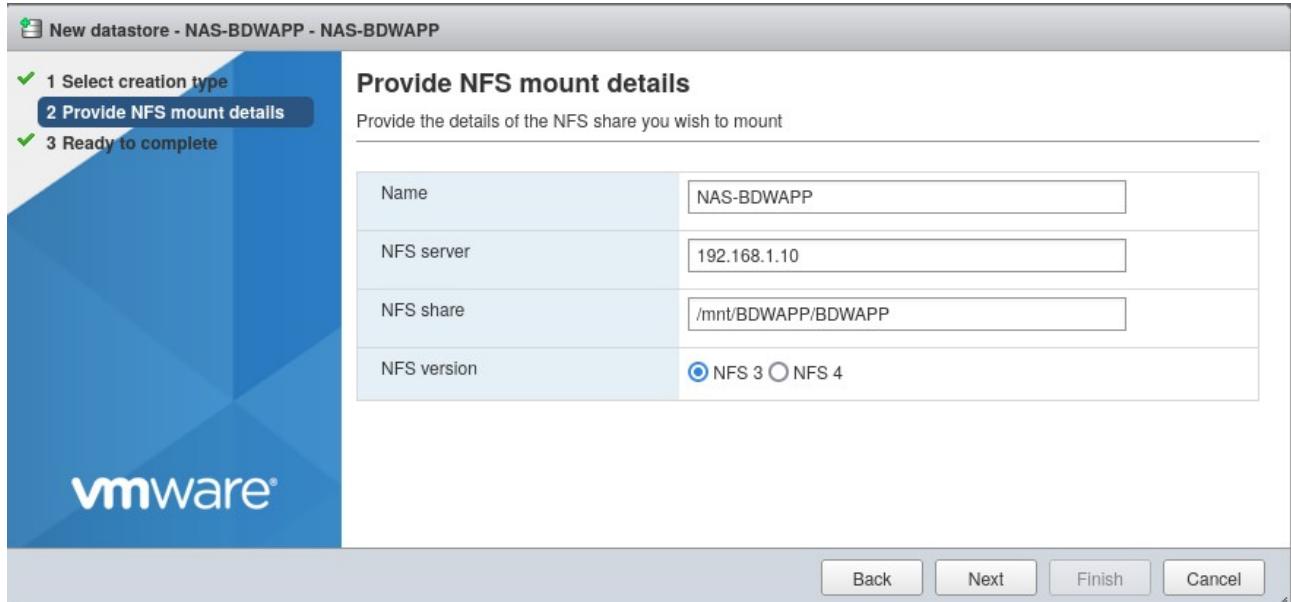
Processo de adição de um Datastore NAS no ESXi

Na tela inicial do ESXi, em datastore. Clicamos em “New datastore”.

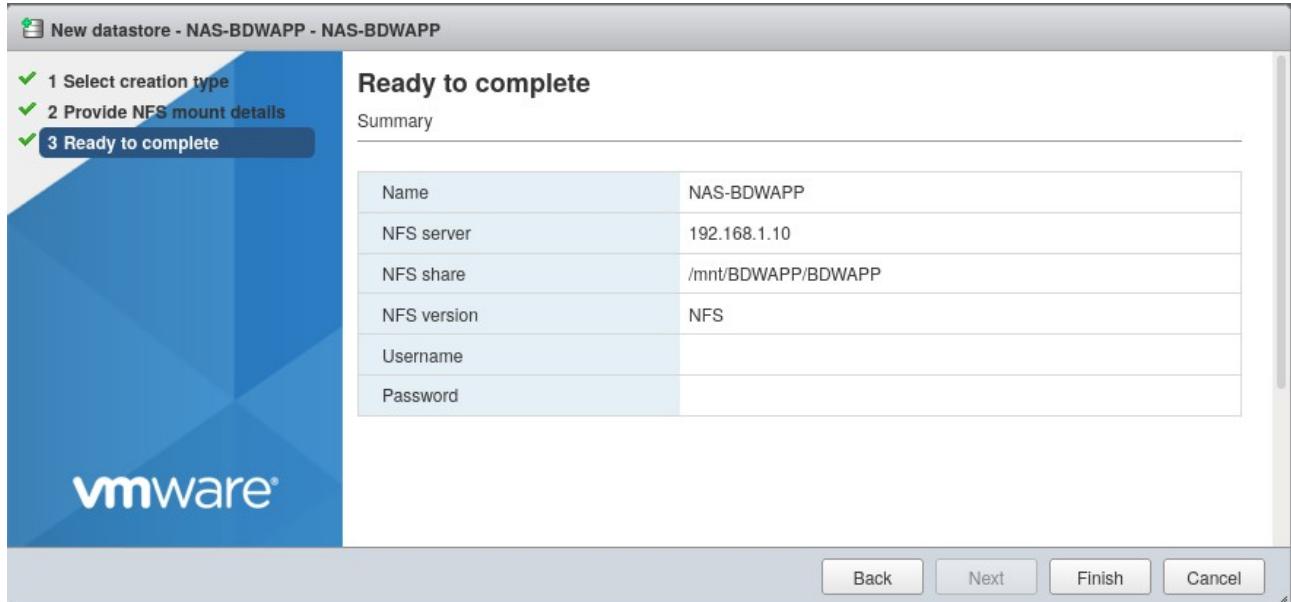
Selecionamos aqui a opção para montar um datastore NFS (**Mount NFS datastore**) e damos um “next”.



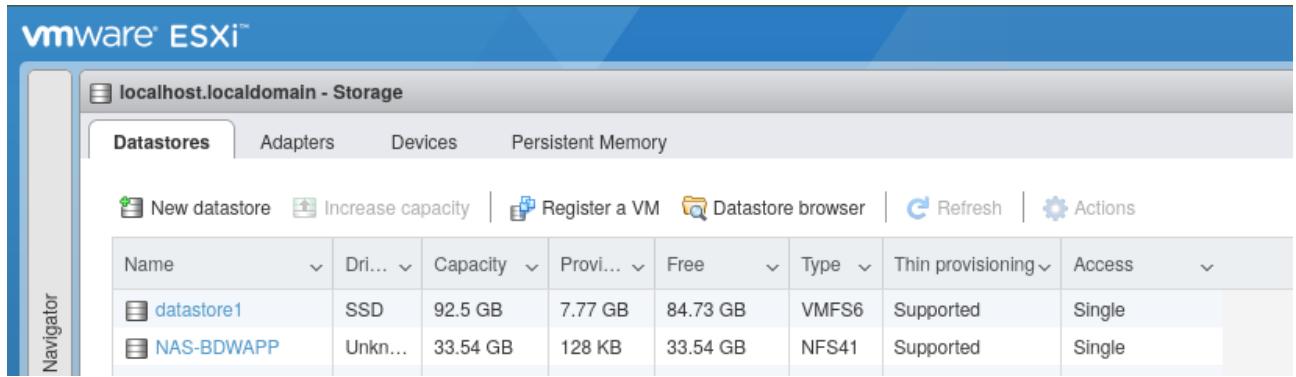
Nesta etapa agora, daremos um nome para esse datastore, preencheremos o campo “**NFS server**” com o IP do servidor NAS criado (freeNAS) e colocaremos o caminho do compartilhamento do storage NAS. E “**next**”.



Confirmamos as informações e “**finish**”.



Pronto! A adição do storage NAS-BDWAPP está concluída.

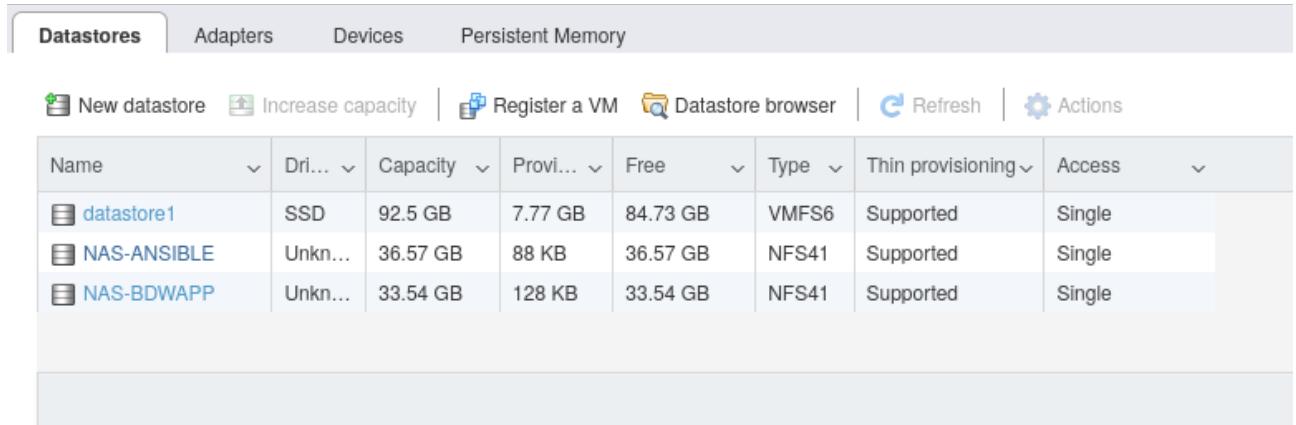


The screenshot shows the VMware ESXi interface under the Storage section. The Datastores tab is selected. There are two datastores listed:

Name	Dri...	Capacity	Provi...	Free	Type	Thin provisioning	Access
datastore1	SSD	92.5 GB	7.77 GB	84.73 GB	VMFS6	Supported	Single
NAS-BDWAPP	Unkn...	33.54 GB	128 KB	33.54 GB	NFS41	Supported	Single

Agora vamos repetir novamente todo o processo para adicionar o segundo volume (NAS-VMWAPP).

Após repetir todos os passos, teremos os dois datastores adicionados e prontos para o uso.



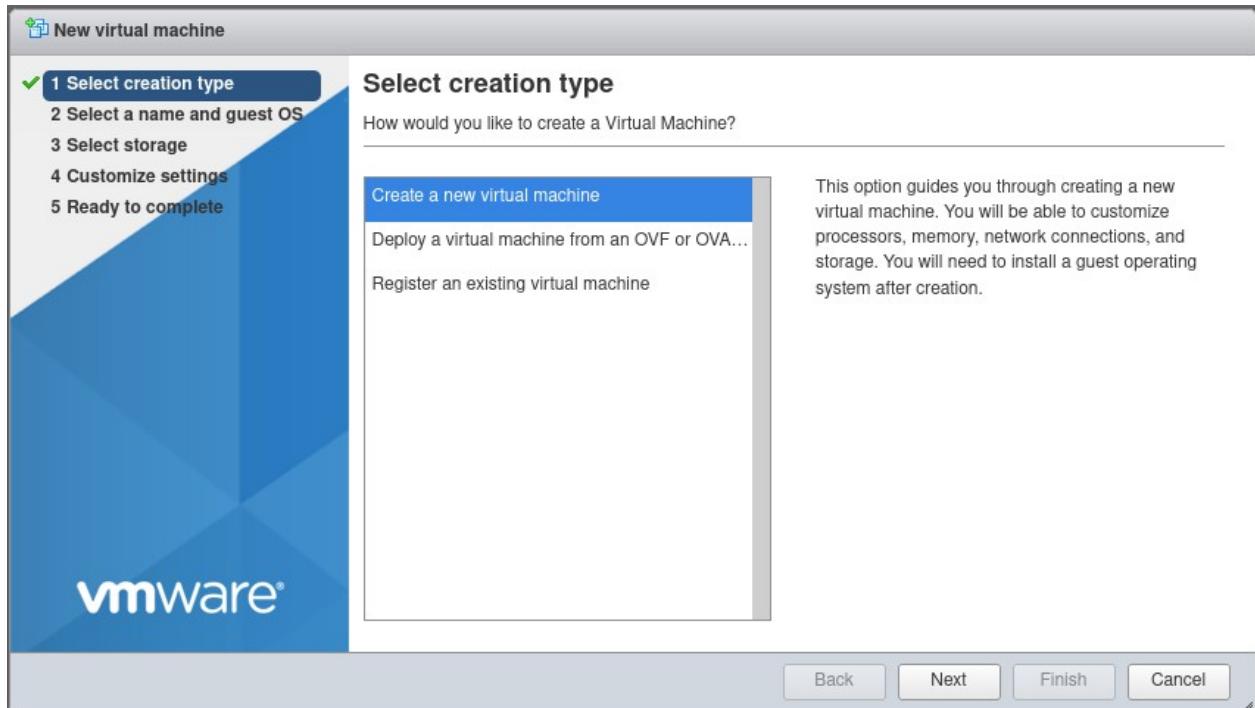
The screenshot shows the VMware ESXi interface under the Storage section. The Datastores tab is selected. There are three datastores listed:

Name	Dri...	Capacity	Provi...	Free	Type	Thin provisioning	Access
datastore1	SSD	92.5 GB	7.77 GB	84.73 GB	VMFS6	Supported	Single
NAS-ANSIBLE	Unkn...	36.57 GB	88 KB	36.57 GB	NFS41	Supported	Single
NAS-BDWAPP	Unkn...	33.54 GB	128 KB	33.54 GB	NFS41	Supported	Single

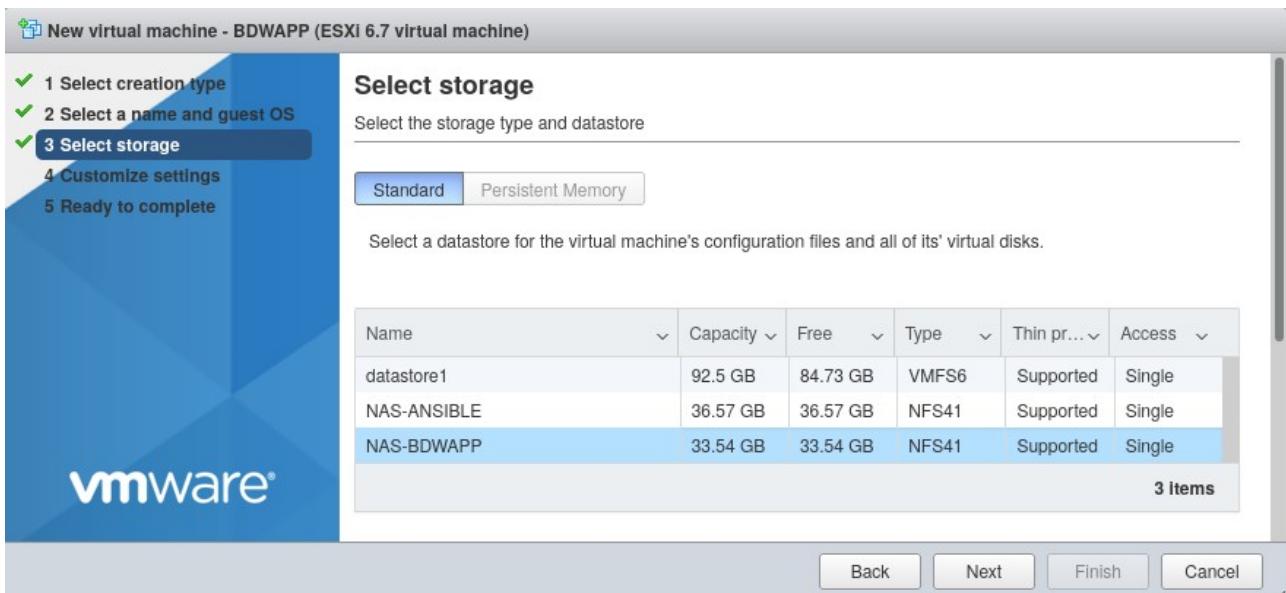
Agora vamos instalar as VMs “**ANSIBLE**” e “**BDWAPP**” nestes novos volumes adicionados.

Processo de criação das VMs usando Datastore Virtual NAS

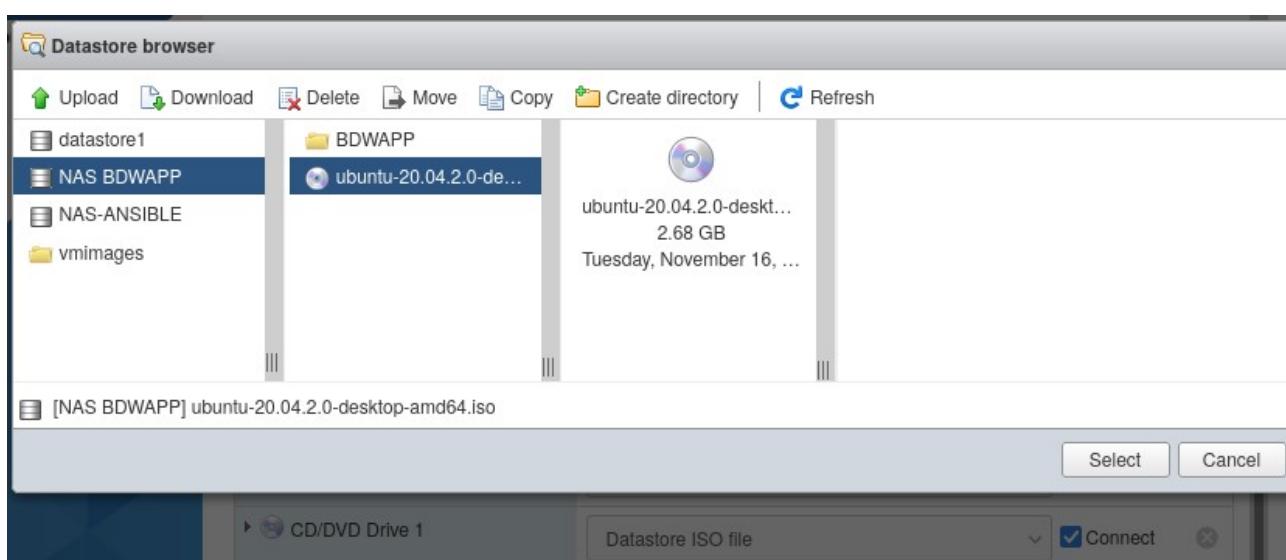
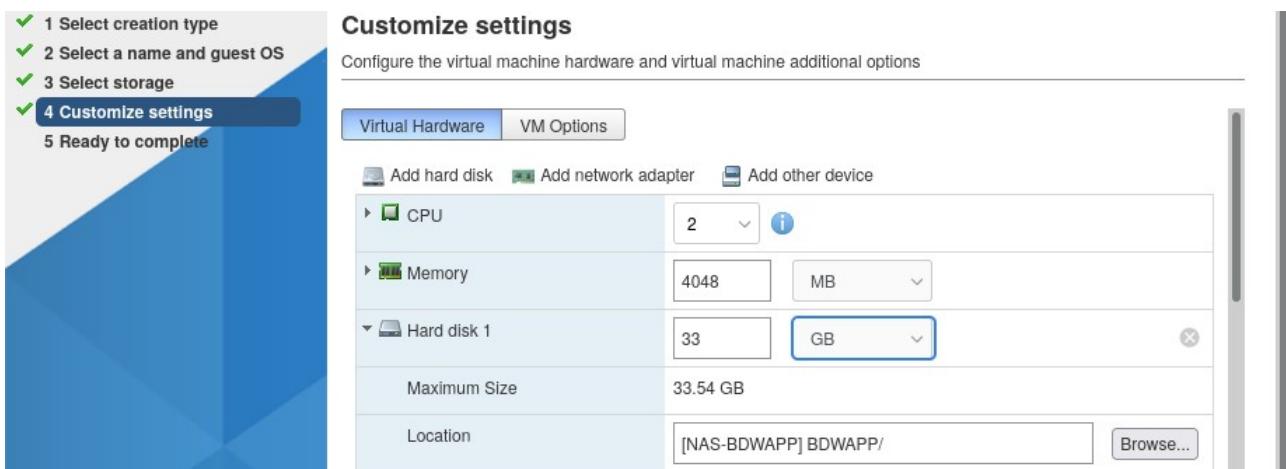
Vamos iniciar instalando a VM **BDWAPP** que será o servidor de automação. O início da instalação segue o padrão normal de qualquer instalação no vmware.



Agora selecionaremos o storage “NAS-BDWAPP” e “next”



Definimos as configurações de hardware, selecionamos a imagem (iso) do sistema operacional a ser instalado e após, de um “Next”.



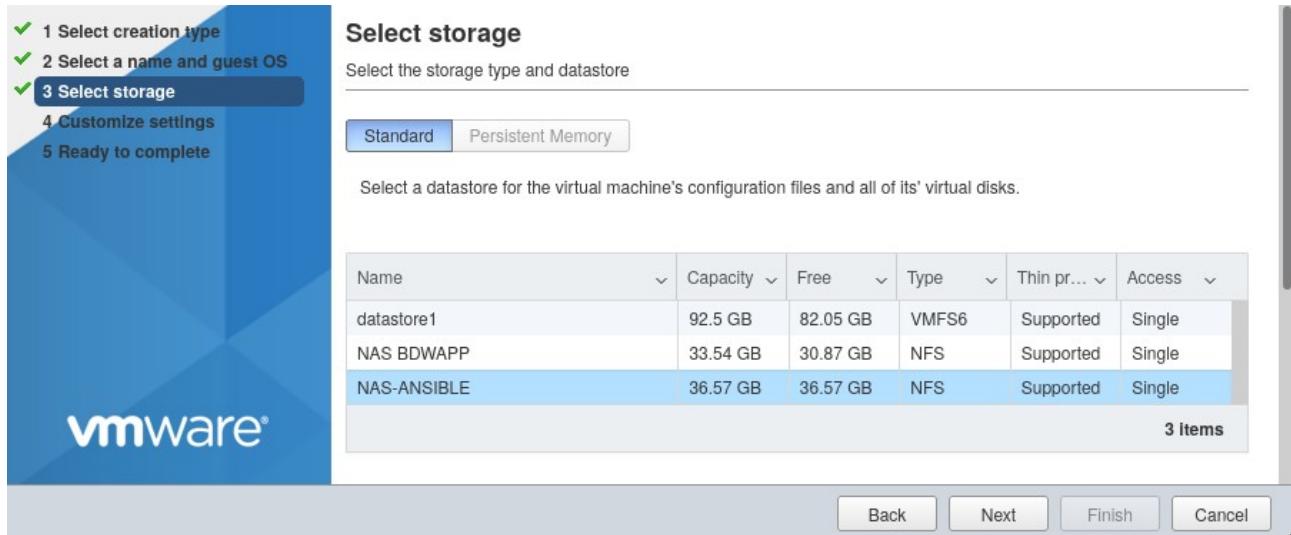
Feito! É só clicar em “Finish”.



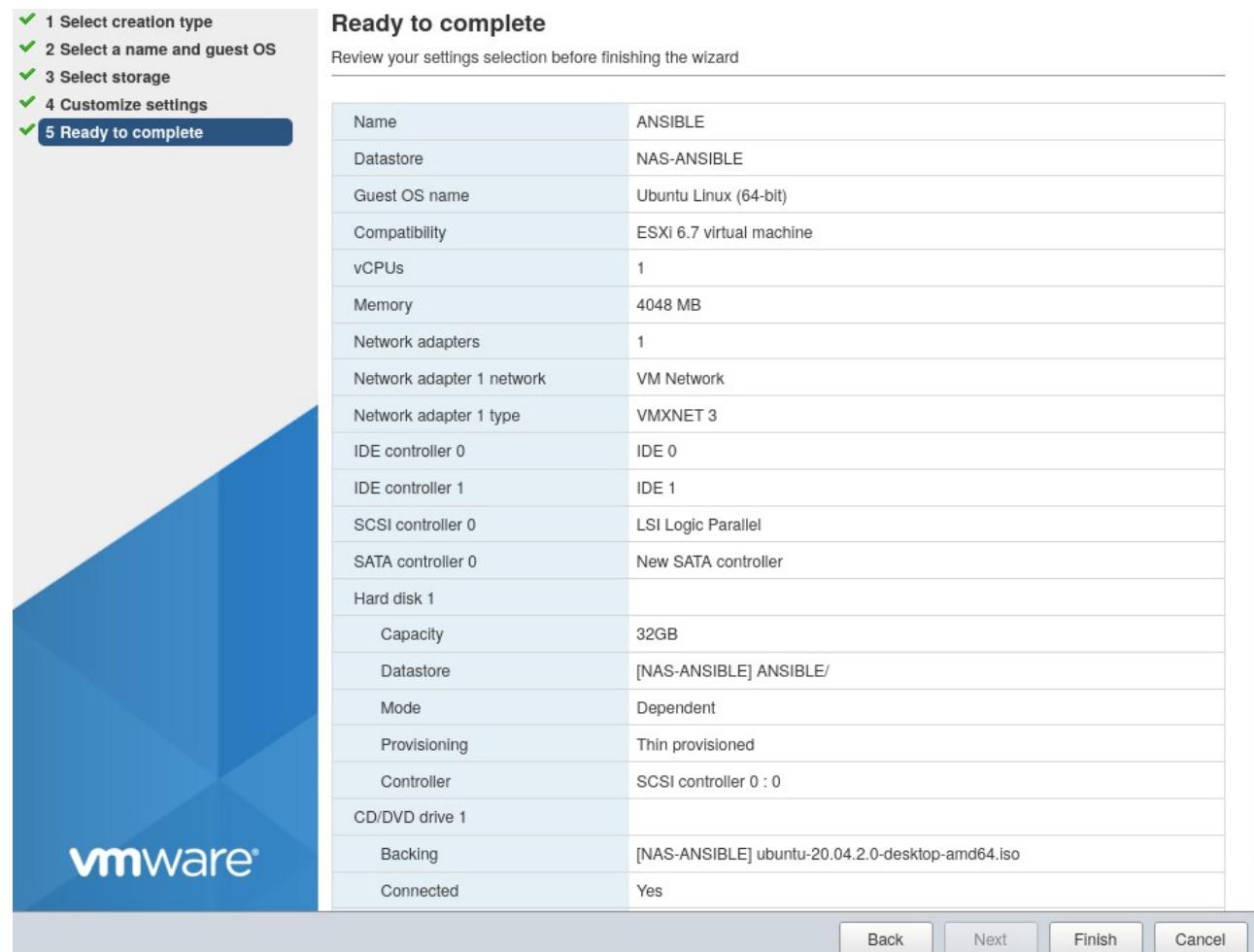
Como podemos ver abaixo a VM já foi criada. Agora basta executar o processo normal de instalação do sistema operacional.



O mesmo processo deve ser feito para criar a VM **ANSIBLE**, mas agora selecionando o storage NAS-ANSIBLE.



Assim como anteriormente, definimos as configurações de hardware, selecionamos a imagem (iso) do sistema operacional a ser instalado e após, de um “**Next**” e “**finish**”.



Como podemos ver abaixo a VM já foi criada. Agora basta executar o processo normal de instalação do sistema operacional.

localhost.localdomain - Virtual Machines

Virtual machine	Status	Used space	Guest OS	Host name	Host CPU	Host memory
BDWAPP	Normal	0 B	Ubuntu Linux (64-bit)	Unknown	0 MHz	0 MB
ANSIBLE	Normal	0 B	Ubuntu Linux (64-bit)	Unknown	0 MHz	0 MB

Informações Gerais das VMs já instaladas “BDWAPP” e “ANSIBLE”

BDWAPP

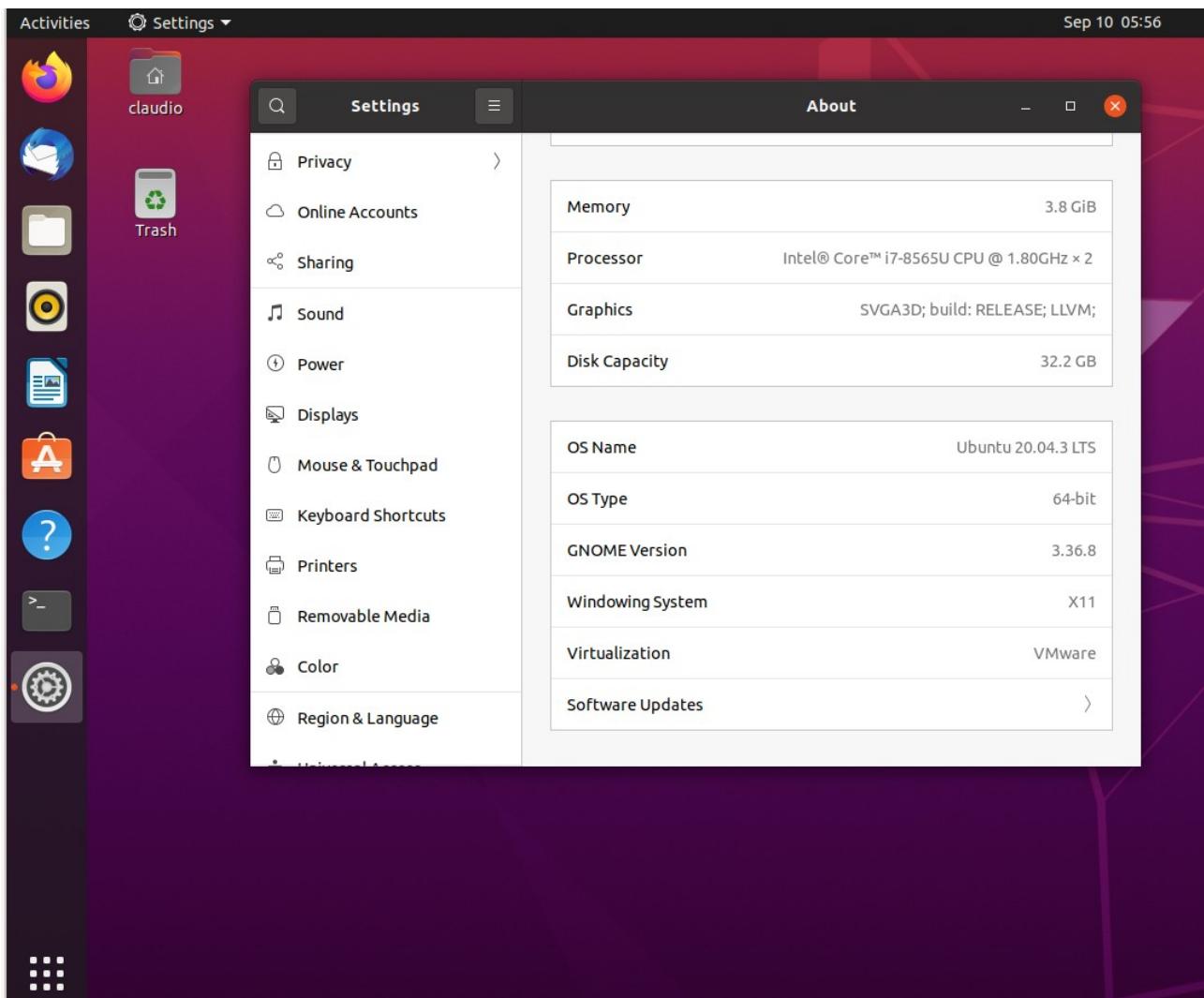
Console	Monitor	Power on	Power off	Suspend	Reset	Refresh	Actions
				BDWAPP Guest OS: Ubuntu Linux (64-bit) Compatibility: ESXi 6.7 virtual machine VMware Tools: No CPUs: 1 Memory: 3.95 GB			
CPU: 79 MHz MEMORY: 4.03 GB STORAGE: 2.96 KB							

ANSIBLE

Console	Monitor	Power on	Power off	Suspend	Reset	Refresh	Actions
				ANSIBLE Guest OS: Ubuntu Linux (64-bit) Compatibility: ESXi 6.7 virtual machine VMware Tools: No CPUs: 1 Memory: 3.95 GB			
CPU: 2.1 GHz MEMORY: 4.01 GB STORAGE: 4.5 KB							

Instalação do Servidor ANSIBLE

Após instalar a VM que será o servidor Ansible. Neste caso, instalamos o ubuntu Its 20.04, vamos preparar o ambiente.



Instalação do Servidor SSH

Após a instalação do Ubuntu e antes de instalar o Ansible, precisaremos instalar o servidor SSH. Já que o ansible utiliza esse protocolo para aplicar configurações nos servidores gerenciados por ele. Basicamente servirá para fazer a conexão entre o ansible e o(s) servidor(es) remoto(s). Além é claro, da natureza segura do SSH.

Para instalar o servidor ssh, basta executar o seguinte comando no terminal do ubuntu.

```
ansible@ansible:~$  
ansible@ansible:~$ sudo apt install openssh-server
```

E após, verificamos o status do servidor para ver se está ativo.

```
ansible@ansible:~$ sudo service ssh status  
● ssh.service - OpenBSD Secure Shell server  
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)  
  Active: active (running) since Thu 2021-11-25 08:45:30 -03; 43s ago  
    Docs: man:sshd(8)  
          man:sshd_config(5)  
  Main PID: 4023 (sshd)  
     Tasks: 1 (limit: 5840)  
   Memory: 1.0M  
  CGroup: /system.slice/ssh.service  
          └─4023 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups  
  
nov 25 08:45:30 ansible systemd[1]: Starting OpenBSD Secure Shell server...  
nov 25 08:45:30 ansible sshd[4023]: Server listening on 0.0.0.0 port 22.  
nov 25 08:45:30 ansible sshd[4023]: Server listening on :: port 22.  
nov 25 08:45:30 ansible systemd[1]: Started OpenBSD Secure Shell server.
```

Feito isso, vamos gerar uma chave para fazer a conexão do ansible com o servidor remoto sem que haja necessidade de estar fazendo o login a todo momento. Para isso, digite o comando: **ssh-keygen** conforme mostra a imagem abaixo e siga confirmando com “enter” até a conclusão.

```
ansible@ansible:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible/.ssh/id_rsa):
Created directory '/home/ansible/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansible/.ssh/id_rsa
Your public key has been saved in /home/ansible/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:L1cOI0tg0hpY9TC+TzyftFYGG3M8qmabtmUz9cqu2+c ansible@ansible
The key's randomart image is:
+---[RSA 3072]---+
| ..+ |
| o o + . |
| . o = . + + |
| = + B . |
| . . S * = |
| + O @ . |
| B % . . |
| o.0 = .. |
| .+.o+=oE |
+---[SHA256]---+
ansible@ansible:~$
```

Agora armazenaremos uma cópia desta chave em um caminho padrão para o serviço SSH (authorized keys) dentro do servidor remoto (BDWAPP).

```
ansible@ansible:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub bdwapp@192.168.1.21
```

Podemos acessar agora nosso servidor sem a necessidade de digitar a senha.

```
ansible@ansible:~$ ssh localhost
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-40-generic x86_64)

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

0 as atualizações podem ser aplicadas imediatamente.

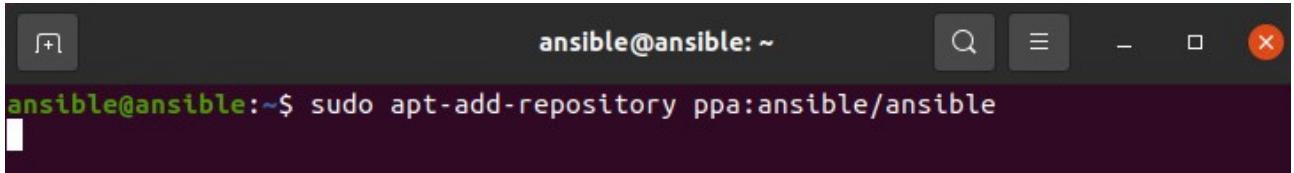
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Thu Nov 25 08:54:47 2021 from 127.0.0.1
ansible@ansible:~$
```

Para sair da sessão SSH basta usar o comando “exit”.

Instalação do Ansible

Para instalar o ansible em nosso sistema gerenciador (ubuntu 20.04) devemos seguir alguns passos:

Primeiro devemos instalar o repositório que se encontra o ansible.



```
ansible@ansible:~$ sudo apt-add-repository ppa:ansible/ansible
```

Observação: A versão do Ubuntu instalada já contempla o python 3.8.10, não sendo necessário a instalação.

```
ansible@ansible:~$ python3 -V
Python 3.8.10
ansible@ansible:~$
```

Agora basta atualizar a lista dos pacotes para ter acesso ao repositório instalado.

```
ansible@ansible:~$ sudo apt update
```

Agora vamos instalar o pacote do ansible

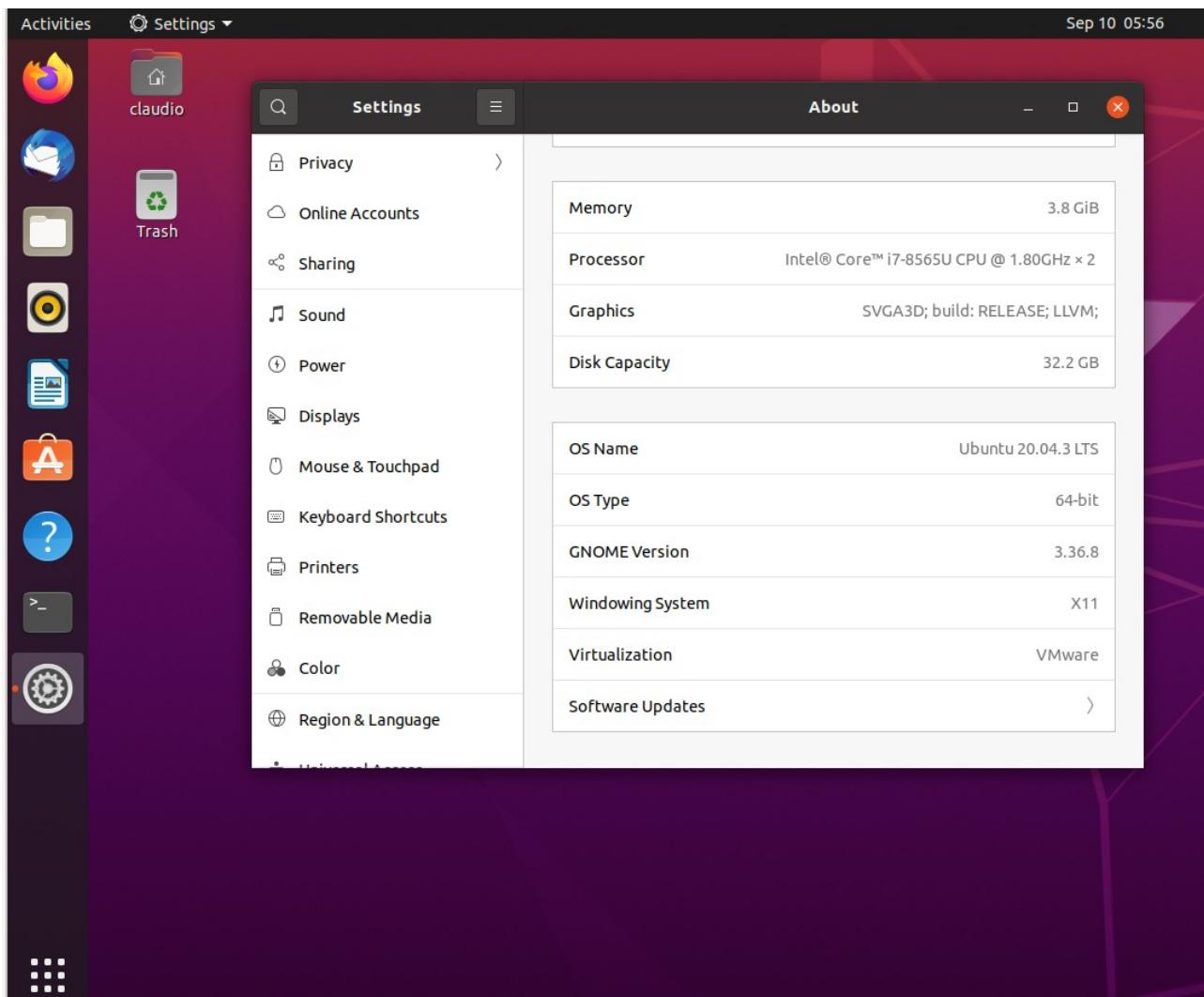
```
ansible@ansible:~$ 
ansible@ansible:~$ sudo apt install ansible
```

Verificando a versão instalada

```
ansible@ansible:~$ ansible --version
ansible [core 2.11.6]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
    ansible python module location = /usr/lib/python3/dist-packages/ansible
    ansible collection location = /home/ansible/.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
```

Instalação do Servidor BDWAPP

Após instalar a VM que será o servidor da Aplicação (Wordpress). Neste caso, instalamos o ubuntu lts 20.04, vamos preparar o ambiente docker.



Preparando o ambiente Docker

Antes de começar a instalação do WordPress, precisamos preparar o ambiente para recebê-lo. Para isso, necessitamos instalar primeiramente o Docker.

Vamos iniciar instalando algumas dependências necessárias para o funcionamento do Docker.

```
bdwapp@bdwapp:~$  
bdwapp@bdwapp:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Por padrão, a versão mais recente do Docker não está disponível no repositório padrão do Ubuntu 20.04, então vamos adicionar o repositório oficial do Docker no sistema.

Mas primeiro, vamos baixar e adicionar a chave GPG para que seja possível autenticar os programas que baixarmos do projeto Docker. Para isso daremos o seguinte comando:

```
bdwapp@bdwapp:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
OK  
bdwapp@bdwapp:~$
```

o comando a seguir exibe a identificação (fingerprint) da chave que acabamos de adicionar.

```
bdwapp@bdwapp:~$ sudo apt-key fingerprint 0EBFCD88  
pub    rsa4096 2017-02-22 [SCEA]  
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88  
uid          [ desconhecida] Docker Release (CE deb) <docker@docker.com>  
sub    rsa4096 2017-02-22 [S]  
bdwapp@bdwapp:~$
```

Em seguida, iremos adicionar o repositório Docker com o seguinte comando:

```
bdwapp@bdwapp:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
Atingido:1 http://br.archive.ubuntu.com/ubuntu focal InRelease
Atingido:2 http://br.archive.ubuntu.com/ubuntu focal-updates InRelease
Atingido:3 http://br.archive.ubuntu.com/ubuntu focal-backports InRelease
Obter:4 https://download.docker.com/linux/ubuntu focal InRelease [57,7 kB]
Atingido:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Obter:6 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [12,9
kB]
Baixados 70,6 kB em 1s (77,5 kB/s)
Lendo listas de pacotes... Pronto
bdwapp@bdwapp:~$
```

Agora vamos utilizar o comando **apt update** para atualizar a lista de pacotes e programas que podem ser instalados na máquina. Incluindo o repositório recém adicionado.

```
bdwapp@bdwapp:~$ sudo apt update
```

Agora sim vamos instalar o docker, conforme o comando abaixo:

```
bdwapp@bdwapp:~$ sudo apt install docker-ce
Lendo listas de pacotes... Pronto
Construindo árvore de dependências
Lendo informação de estado... Pronto
Os pacotes adicionais seguintes serão instalados:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin git
    git-man liberror-perl pigz slirp4netns
Pacotes sugeridos:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit
    git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
Os NOVOS pacotes a seguir serão instalados:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras
    docker-scan-plugin git git-man liberror-perl pigz slirp4netns
0 pacotes atualizados, 10 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 101 MB de arquivos.
Depois desta operação, 441 MB adicionais de espaço em disco serão usados.
Você quer continuar? [S/n]
```

Pronto! Com todo o ambiente Ansible e Docker já preparados, vamos iniciar o processo de criação de um playbook para rodar o WordPress.

```
bdwapp@bdwapp:~$ docker --version
Docker version 20.10.11, build dea9396
bdwapp@bdwapp:~$
```

Preparando um Playbook para containers

Voltaremos novamente a **VM ANSIBLE** para iniciarmos o processo de automação.

Para melhor organizar a instalação e facilitar o gerenciamento de todo o código, vamos criar uma pasta. Para isso usaremos o comando “**mkdir**”.

```
ansible@ansible:~$ mkdir wordpress-docker  
ansible@ansible:~$
```

Após criado o diretório “**wordpress-docker**”, precisamos criar dois arquivos dentro dele. Um que vai conter nosso playbook e outro para armazenar os hosts (que será usado para apontar o servidor), que é o nosso inventário. Para isso usaremos o comando “**touch**”.

```
ansible@ansible:~$ cd wordpress-docker/  
ansible@ansible:~/wordpress-docker$ touch hosts  
ansible@ansible:~/wordpress-docker$ touch playbook.yml
```

Escrevendo o arquivo de inventário (hosts)

Vamos editar o arquivo de inventário para colocar o endereço IP da máquina onde pretendemos fazer a instalação. Para isso abriremos o arquivo “**hosts**” com o editor “**nano**”.

```
ansible@ansible:~/wordpress-docker$ nano hosts  
ansible@ansible:~/wordpress-docker$
```

Após darmos um nome e configurar o IP (**VM BDWAPP**), daremos o comando “**ctrl+o**” para salvar e “**enter**” para confirmar o nome do arquivo e após, é só dar comando “**ctrl+x**” para sair.

```
GNU nano 4.8                               hosts  
[WP-DOCKER]  
192.168.1.21  
  
^G Obter Ajuda  ^O Gravar      ^W Onde está?  ^K Recort txt  ^J Justificar  
^X Sair        ^R Ler o arq  ^V Substituir  ^U Colar txt  ^T Verf0rtog
```

Pronto! Agora vamos escrever o playbook.

Criando o Playbook

Antes de iniciar, vamos modificar a configuração do arquivo “**sudoers**”, para que possamos executar comandos no Linux como um usuário sudo sem ter que inserir a senha todas as vezes.

```
ansible@ansible:~$ sudo nano /etc/sudoers
```

Deixe a linha conforme abaixo:

```
#includedir /etc/sudoers.d
ansible ALL=NOPASSWD: ALL
#
^G Obter Ajuda ^O Gravar      ^W Onde está? ^K Recort txt  ^J Justificar  ^C Pos atual M-U
^X Sair        ^R Ler o arq  ^\ Substituir ^U Colar txt   ^T VerfOrtog   ^ I Ir p/ linha M-E
```

Agora sim! Da mesma forma que fizemos para o arquivo de inventário, vamos editar o arquivo playbook.yml através do editor “nano” e salvar.

```
ansible@ansible:~$ cd wordpress-docker/
ansible@ansible:~/wordpress-docker$ nano playbook.yml
```

```
GNU nano 4.8                               playbook.yml
---
- hosts: WP-DOCKER
  remote_user: bdwapp
  become: yes
  tasks:
    - name: "Executa o container MySQL"
      docker_container:
        name: bdwordpress
        image: mysql:5.7
        env:
          MYSQL_ROOT_PASSWORD: Password123
          MYSQL_DATABASE: wordpress
          MYSQL_USER: wordpress
          MYSQL_PASSWORD: Password123

    - name: "Executa o container WordPress"
      docker_container:
        name: wordpress
        image: wordpress
        env:
          WORDPRESS_DB_HOST: bdwordpress:3306
          WORDPRESS_DB_USER: wordpress
          WORDPRESS_DB_PASSWORD: Password123
        links:
          - "bdwordpress:mysql"
        ports:
          - "80:80"
```

Antes de validar o playbook, precisaremos instalar para esta versão do Ubuntu e do Docker que estamos utilizando, 2 componentes adicionais de Python para que o Docker funcione de forma automatizada via Ansible.

```
ansible@ansible:~/wordpress-docker$ sudo apt install python3-pip
[...]
ansible@ansible:~/wordpress-docker$ sudo pip3 install docker
```

Agora um último passo é necessário antes de validarmos o playbook. Vamos voltar até o servidor **BDWAPP** e modificar o arquivo [sudoers conforme já feito](#) para o servidor **ANSIBLE**. Mas agora adicionando o usuário “**bdwapp**”

Validando o Playbook

Agora faremos a validação do playbook. Como pode ser visto na imagem abaixo tudo ocorreu perfeitamente.

```
ansible@ansible:~/wordpress-docker$ ansible-playbook -i hosts playbook.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [WP-DOCKER] ****
TASK [Gathering Facts] ****
ok: [192.168.1.21]

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: [192.168.1.21]

TASK [Executa o container WordPress] ****
changed: [192.168.1.21]

PLAY RECAP ****
192.168.1.21 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Agora se formos até o servidor **BDWAPP**, podemos verificar na imagem abaixo que os containers criados: “**banco**” e “**meuwordpress**” já estão em execução.

```
bdwapp@bdwapp:~$ sudo docker ps
CONTAINER ID   IMAGE       COMMAND           CREATED      STATUS      PORTS          NAMES
d6c48151a59c   wordpress   "docker-entrypoint.s..."  6 minutes ago   Up 6 minutes   0.0.0.0:80->80/tcp   wordpress
5e25fa29f9d6   mysql:5.7  "docker-entrypoint.s..."  7 minutes ago   Up 7 minutes   3306/tcp, 33060/tcp   banco
bdwapp@bdwapp:~$
```

Acessando o WordPress pelo navegador

Terminado a instalação através do playbook criado na **VM ANSIBLE**, agora é só acessar através de um navegador web. Para isso vamos digitar o endereço de IP da **VM BDWAPP** na barra de endereços do navegador.

seguro | 192.168.1.21/wp-admin/install.php?step=1



The screenshot shows the first step of the WordPress installation process. At the top, there's a large blue 'W' logo. Below it, the heading 'Bem-vindo (a)' is displayed. A sub-headline reads: 'Bem-vindo (a) à famosa instalação do WordPress em cinco minutos! Basta preencher as informações abaixo e você estará a poucos passos de usar a plataforma de publicação mais extensível e poderosa do mundo.' Underneath, a section titled 'Informação necessária' asks for site title and administrator username. The site title is 'LOGICOL - Ponto Eletrônico' and the username is 'wordpress'. A note below says: 'Nomes de usuário podem ter somente caracteres alfanuméricos, espaços, sublinhados, hífens, pontos e o símbolo @.' The URL in the address bar is '192.168.1.21/wp-admin/install.php?step=1'.



The screenshot shows the WordPress dashboard. The top navigation bar includes links for Personalizar, Novo, and Editar página. On the right, there are user profile and search icons. The main content area features a logo consisting of three overlapping curved bands in blue, green, and orange, followed by the text 'Registro de Ponto online'. The URL in the address bar is '192.168.1.21'.

Resgite seu Ponto de forma rápida e segura

Preparando o ambiente GIT local

Agora iremos armazenar todo esse nosso código gerado no sistema de controle de versão da GitHub para que possamos registrar alterações nos arquivos ao longo do tempo. Primeiramente iniciaremos instalando o GIT no servidor ansible.

```
ansible@ansible:~/wordpress-docker$ sudo apt install git
```

Agora devemos entrar dentro do diretório “**wordpress-docker**” para prepará-lo para a operação com o GIT. Usaremos o comando “**git init**”.

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

Vamos adicionar toda a estrutura do projeto ao sistema de controle de versão local.

```
ansible@ansible:~/wordpress-docker$ git add *
ansible@ansible:~/wordpress-docker$
```

Antes de fazermos primeiro commit, precisamos nos autenticar no GIT para que ele possa aceitar o projeto. Lembrando que essa é uma identificação local.

Faremos conforme a imagem abaixo.

```
ansible@ansible:~/wordpress-docker$ git config --global user.email "claudio.grande@al.infnet.edu.br"
ansible@ansible:~/wordpress-docker$ git config --global user.name "cazgr"
```

Fazendo um Commit

Vamos fazer nosso primeiro “commit”.

```
ansible@ansible:~/wordpress-docker$ git commit -m "importando playbook ansible docker wordpress"
[master (root-commit) fa6d714] importando playbook ansible docker wordpress
 2 files changed, 29 insertions(+)
  create mode 100644 hosts
  create mode 100644 playbook.yml
ansible@ansible:~/wordpress-docker$
```

Observação: O commit dos instantâneos do Git sempre é feito no repositório local. O parâmetro “**-m**” seguido do texto entre aspas, é referente a mensagem que queremos enviar junto ao commit. Basicamente informar o que este commit faz.

Vamos agora apontar o projeto que acabamos de criar para o nosso repositório GitHub:

```
ansible@ansible:~/wordpress-docker$ git remote add origin https://github.com/cazgr/WP-Docker-Ansible.git
ansible@ansible:~/wordpress-docker$
```

Como pode ser visto na consulta abaixo, o repositório remoto foi apontado corretamente.

```
ansible@ansible:~/wordpress-docker$ 
ansible@ansible:~/wordpress-docker$ git remote -v
origin  https://github.com/cazgr/WP-Docker-Ansible.git (fetch)
origin  https://github.com/cazgr/WP-Docker-Ansible.git (push)
ansible@ansible:~/wordpress-docker$
```

Fazendo um Push

Dentro do terminal, acessaremos a pasta **wordpress-docker** e daremos o comando **git push** apontando para o branch master.

Após executarmos o comando, nos será solicitado o nosso nome de usuário e a senha (token) para que possamos nos autenticar no GitHub.

```

ansible@ansible:~/wordpress-docker$ git push -f origin master
Username for 'https://github.com': cazgr
Password for 'https://cazgr@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), 577 bytes | 577.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/cazgr/WP-Docker-Ansible.git
 + 9c8d791...fa6d714 master -> master (forced update)
ansible@ansible:~/wordpress-docker$
```

Procedimento feito com sucesso! Toda a nossa estrutura criada localmente (**wordpress-docker**) foi para o servidor git remoto. Como pode ser visto na imagem abaixo, agora nosso repositório no Github está populado com as configurações que fizemos na nossa infraestrutura local.

The screenshot shows a GitHub repository page for 'cazgr / WP-Docker-Ansible'. The repository is public and has 1 pull request, 0 stars, and 0 forks. The code tab is selected. A commit from 'cazgr' titled 'importando playbook ansible docker wordpress' was made 1 hour ago. Below the commit, there are links for 'hosts' and 'playbook.yml'. A note at the bottom encourages adding a README. The repository has no description, website, or topics provided.

Todo o código pode ser acessado pelo link: <https://github.com/cazgr/WP-Docker-Ansible>

Estimativa de Custos

Relação dos Dispositivos Finais e Intermediários de Rede	Quantidade	Valor Unit.	Valor Total	Local de Consulta
Switch Cisco SG350 24 Portas Gigabit 24x PoE 382W 2x SFP 2x Combo Mini-Gbic Gerenciável Layer3 MPN: SG350-28MP-K9-BR	2	R\$ 7.861,65	R\$ 15.723,30	https://www.fourserv.com.br/switch-cisco-28-portas-28x-10-100-1000-gerenciavel
Firewall Netgate XG-7100 1U	1	\$999,00	\$999,00	https://shop.netgate.com/products/7100-base-pfsense
Servidor de Rack PowerEdge R740 Intel Xeon Gold 5215 (2.5 GHz, 10 núcleos/20 threads, 10.4GT/s, 13.75M Cache, Turbo, TDP 85W DDR4-2666)	2	R\$ 61.691,00	R\$ 123.382,00	https://www.dell.com/pt-br/work/shop/servidores-armazenamento-rede/servidor
Rack Fechado 44U – RC19 Standard 700mm	1	R\$ 2.878,38	R\$ 2.878,38	https://shoppingdosracks.com.br/loja/rack-servidor-fechado-44u/
NAS Qnap TS-1673AU-RP 16 Baias CPU Quad-Core 2,2 GHz 16GB DDR4 2 Portas 2.5GbE Sem discos	1	R\$ 50.149,15	R\$ 50.149,15	https://www.fourserv.com.br/nas-qnap-ts-1673au-rp-16-baias-cpu-quad-core-2-2-ghz
HD Interno WD RED 10TB Ideal para NAS 24x7 MPN: WD101EFAX	8	R\$ 2.289,15	R\$ 18.313,20	https://www.fourserv.com.br/hd-interno-wd-red-10tb-ideal-para-nas-mpn-wd101efax
SSD WD RED SA500 NAS 500GB SATA 6Gb/s Ideal para NAS MPN: WDS500G1R0A	2	R\$ 1.002,15	R\$ 2.004,30	https://www.fourserv.com.br/ssd-wd-red-sa500-nas-500gb-sata-6gb-s-ideal-para-nas
HD Interno WD RED 1TB Ideal para NAS 24x7 MPN: WD10EFRX	4	R\$ 619,65	R\$ 2.478,60	https://www.fourserv.com.br/hd-interno-wd-red-1tb-ideal-para-nas-mpn-wd10efrx

Observações:

\$ Preços em Dólar – Cotação apenas em dólares, portanto o preço final pode variar a depender da cotação do dia.

R\$ Preços em Reais

Cronograma

Projeto de Arquitetura de Infraestrutura da Aplicação Ponto Remoto

Etapa	Ação	Status	Previsto	Realizado
Etapa 1 – Levantamento de Requisitos	Definição dos Objetivos do Projeto	Feito	15. ago	15. ago
	Pilha de Softwares da aplicação	Feito	2. set	2. set
	Validação dos Requisitos	Feito	4. set	4. set
Etapa 2 – Escolha da Solução	Apresentação das possíveis soluções de infraestrutura	Feito	19. nov	18. nov
	Validação da solução escolhida	Feito	19. nov	18. nov
	Apresentação das tecnologias que compõe a solução escolhida	Feito	19. nov	18. nov
Etapa 3 – Infraestrutura	Apresentação da Infraestrutura To Be	Feito	19. nov	19. nov
	Diagrama de Infraestrutura To Be	Feito	19. nov	2. nov
	Apresentação da Infraestrutura As Is	Feito	19. nov	19. nov
	Diagrama de Infraestrutura As Is	Feito	19. nov	2. nov
Etapa 4 - Modelo de Simulação infraestrutura	Instalação VMware Workstation (infraestrutura física simulada)	Feito	5. set	30. ago
	Instalação VM FreeNAS	Feito	19. nov	12. nov
	Configuração dos volumes	Feito	19. nov	12. nov
	Instalação ESXi	Feito	5. set	2. set
	Configuração ESXi	Feito	5. set	2. set
	Adição dos Datastore NAS	Feito	19. set	12. nov
	Instalação da VM Servidor BDWAPP	Feito	19. nov	24. nov
	Instalação dos requisitos e configuração do ambiente	Feito	19. nov	24. nov
	Testes de Validação	Feito	19. nov	24. nov
	Instalação da VM Servidor ANSIBLE	Feito	19. nov	24. nov
	Instalação dos requisitos e configuração do ambiente	Feito	19. nov	24. nov
	Testes de Validação	Feito	19. nov	24. nov
	Teste final de validação do ambiente simulado	Em progresso	28. nov	

Etapa 5 – Entrega Documentação		Status	Previsto	Realizado
Solução Proposta		Feito	30. nov	29. out
Projeto Infraestrutura To Be		Feito	30. nov	19. nov
Requisitos da infraestrutura		Feito	30. nov	8. nov
Apresentação detalhada da Infraestrutura		Feito	30. nov	19. nov
Diagrama de Infraestrutura física e lógica		Feito	30. nov	12. nov
Projeto Infraestrutura As Is		Feito	30. nov	19. nov
Requisitos de hardware do ambiente simulado		Feito	30. nov	9. nov
Diagrama de Infraestrutura física e lógica		Feito	30. nov	12. nov
Processo de instalação do modelo de simulação		Feito	30. nov	19. nov
Revisão da documentação		Em progresso	28. nov	

Etapa 6 – Apresentação		Status	Previsto	Realizado
Apresentação do Projeto		Parado	7. dez	
Produção do Slide de apresentação		Parado	3. dez	
Gravação do Vídeo		Parado	6. dez	

Referências Bibliográficas

INFNET. Roteiro de Aprendizagem, 2021. O modelo de Computação-Armazenamento-Comunicação. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253291>>. Acesso em: 25 de julho de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Funcionalidades para um SDDC. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253297>>. Acesso em: 3 de agosto de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Instalando e configurando um host ESXi. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253304>>. Acesso em: 10 de agosto de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Instalando e configurando um servidor vCenter. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253310>>. Acesso em: 3 de agosto de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Gerenciar armazenamento iSCSI com VMware ESXi. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253316>>. Acesso em: 25 de julho de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Gerenciar armazenamento compartilhado com VMware ESXi. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253322>>. Acesso em: 3 de agosto de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Gerenciar redes virtuais com o VMware ESXi. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253329>>. Acesso em: 02 de setembro de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Gerenciar backups, snapshots e migração de máquinas virtuais. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253335>>. Acesso em: 03 de setembro de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Implementar virtualização de servidores via containers de aplicações. Disponível em:
<<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253342>>. Acesso em: 03 de setembro de 2021.

SENIORMEGA. Requisitos técnicos - Mega Planier - Ambientes On Premises (software homologado). Disponível em:

<<https://taskcenter.mega.com.br/hc/pt-br/articles/360003643933-Requisitos-t%C3%A9cnicos-Mega-Planier-Ambientes-On-Premises-software-homologado->>. Acesso em: 17 de setembro de 2021.

VMWARE. Requisitos de hardware, 2021. ESXi Hardware Requirements. Disponível em: <<https://docs.vmware.com/en/VMware-vSphere/6.7/com.vmware.esxi.upgrade.doc/GUID-DEB8086A-306B-4239-BF76-E354679202FC.html>>. Acesso em: 18 de setembro de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Propor soluções de arquitetura de infraestrutura para aplicações de acordo com o cenário. Disponível em:

<<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253088>>. Acesso em: 03 de setembro de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Utilizar ferramentas de automação de instalação de aplicações. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253096>>. Acesso em: 31 de julho de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Documentar um projeto de infraestrutura para aplicações. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253104>>. Acesso em: 31 de julho de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Selecionar a plataforma de execução para uma aplicação. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253111>>. Acesso em: 27 de setembro de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Documentando sua infraestrutura de nuvem. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253126>>. Acesso em: 09 de novembro de 2021.

INFNET. Roteiro de Aprendizagem, 2021. Mapeando nossa aplicação em containers Docker. Disponível em: <<https://lms.infnet.edu.br/moodle/mod/page/view.php?id=253134>>. Acesso em: 11 de novembro de 2021.