



Disponibilidade e Desempenho

High Availability Cluster em Proxmox

Pedro Miguel Neves Martins
2021135054(@isec.pt)

Miguel Umbelino da Mota Roma
2021138955(@isec.pt)

Licenciatura em Engenharia Informática
Ramo de Redes e Administração de Sistemas
Instituto Superior de Engenharia de Coimbra
Instituto Politécnico de Coimbra

Coimbra, 5 de janeiro 2024

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	1
1.3	Estrutura do Relatório	2
2	Conceitos	3
2.1	<i>Cluster</i>	3
2.2	<i>High Availability</i>	3
2.3	<i>Nested Virtualization</i>	3
2.4	<i>Quorum disk</i>	4
3	Tecnologias Usadas	5
3.1	<i>Proxmox</i>	5
3.2	<i>High-Availability Cluster</i>	5
3.3	<i>Ceph</i>	6
4	Planeamento	7
4.1	Ambiente laboratorial	7
4.2	Experiências	8
5	Guião Laboratorial	9
5.1	Instalações necessárias	9
5.2	VirtualBox	9
5.3	Proxmox	10
5.3.1	Criação da <i>Virtual Machine</i>	10
5.3.2	Instalação do <i>Proxmox</i>	11
5.4	Configuração do ficheiro <i>hosts</i>	14
5.5	Criação de um <i>Cluster</i>	14
5.6	Gestão de utilizadores	16
5.6.1	Criação do utilizador <i>PAM</i>	17
5.6.2	Criação do utilizador <i>Proxmox</i>	18

5.6.3	Criação de um grupo	18
5.7	Rede separada do <i>Cluster</i>	20
5.8	Criação de uma máquina Virtual	21
5.9	<i>Ceph</i>	22
5.9.1	Instalação nos nós	23
5.9.2	Criação do <i>OSD</i>	23
5.9.3	Criação de uma <i>Pool</i>	24
5.9.4	Configuração do monitor	24
5.10	Criação de um <i>Container</i>	24
5.11	<i>High-Availability group</i>	26
5.12	<i>Node failure</i>	28
5.12.1	Adição de <i>High-Availability</i>	28
5.12.2	Validação <i>storage</i>	29
6	Conclusões	31
6.1	Objetivos alcançados	31
6.2	Problemas encontrados	31
6.3	Problemas em aberto	31
6.4	Opinião acerca da tecnologia estudada	32
	Referências	33

Lista de Figuras

4.1	Topologia experimental	7
5.1	Instalação <i>Virtual Machine</i> (VM) <i>Proxmox</i>	10
5.2	Configurações finais da VM	11
5.3	Ativação da <i>Nested Virtualization</i>	11
5.4	Configuração da placa de rede da VM	11
5.5	Obtenção de um <i>Internet Protocol</i> (IP) através de <i>Dynamic Host Configuration Protocol</i> (DHCP)	12
5.6	Seleção do disco rígido	12
5.7	Configuração do fuso horário e localização	12
5.8	Definição de <i>password</i> e <i>email</i>	13
5.9	Configurações da rede	13
5.10	Remoção do ficheiro <i>International Organization for Standardization</i> (ISO)	13
5.11	Adicionar servidores ao ficheiro <i>hosts</i>	14
5.12	Criação do <i>Cluster</i>	15
5.13	Copia da informação do <i>cluster</i>	15
5.14	Adesão ao <i>Cluster</i>	16
5.15	Interface do nó <i>PVE02</i>	16
5.16	Criação utilizador <i>Pluggable Authenticoitin Module</i> (PAM)	17
5.17	Utilizador PAM não existe	17
5.18	Criação do utilizador PAM através da <i>shell</i>	18
5.19	Criação do utilizador <i>Proxmox</i>	18
5.20	Criação de um grupo	19
5.21	Permissões do grupo	19
5.22	Permissões do utilizador	19
5.23	Vista do perfil do utilizador	20
5.24	Adicionar a segunda placa aos servidores	20
5.25	Configuração da segunda interface	20
5.26	Conectividade entre os servidores	21
5.27	Seleção da ISO a usar	21
5.28	Criação de uma Máquina Virtual	21

5.29 Criação de uma Máquina Virtual	22
5.30 Configuração do Kali	22
5.31 <i>Kali Linux</i>	22
5.32 Escolha de instalação do <i>Ceph</i>	23
5.33 Escolha do novo disco criado	23
5.34 Configuração <i>Object Storage Daemons</i> (OSD) final	24
5.35 Criação da <i>Ceph pool</i>	24
5.36 Monitores criados no <i>Ceph</i>	24
5.37 Instalação de um <i>template</i>	25
5.38 Criação de um <i>container</i>	25
5.39 Associação do <i>template</i> ao <i>container</i>	26
5.40 <i>Shell</i> do <i>container</i>	26
5.41 Criação de um grupo <i>High Availability</i> (HA)	27
5.42 Adição de um <i>container</i> ao grupo	27
5.43 Adição de uma VM ao grupo	27
5.44 <i>Resources</i>	28
5.45 <i>Resources HA</i>	28
5.46 Escolha do storage na VM a migrar	29
5.47 Configuração pré-falha no <i>PVE01</i>	29
5.48 Configuração pós-falha no <i>PVE01</i>	30

Acrónimos e Siglas

API *Application Programming Interface*

CA *Certificate Authority*

DHCP *Dynamic Host Configuration Protocol*

GUI *Graphical User Interface*

HA *High Availability*

IP *Internet Protocol*

ISO *International Organization for Standardization*

NAT *Network Address Translation*

NTP *Network Time Protocol*

OSD *Object Storage Daemons*

PAM *Pluggable Authenticoiin Module*

RAM *Random Access Memory*

SSH *Secure Shell*

VM *Virtual Machine*

Capítulo 1

Introdução

Neste relatório encontra-se documentado todo o trabalho desenvolvido ao longo do primeiro semestre do ano letivo de 2023/2024, no âmbito do Projeto B da unidade curricular Disponibilidade e Desempenho da licenciatura em Engenharia Informática - Ramo de Redes e Administração de Sistemas do Instituto Superior de Engenharia de Coimbra.

1.1 Enquadramento

A ideia de *clusters* de alta disponibilidade não é nova, mas a tecnologia e as ferramentas evoluíram de tal maneira que, atualmente, a sociedade encontra-se dependente das informações fornecidas pelos computadores através da rede. Os dispositivos móveis ampliaram essa dependência, ao dar a possibilidade de aceder à rede a qualquer momento e em qualquer lugar, tornando-se crucial assegurar que os serviços necessários estejam sempre operacionais e sejam possíveis de aceder.

1.2 Objetivos

O projeto aqui relatado teve como objetivo o estudo, em termos teóricos e experimentais, de uma tecnologia na perspetiva da presente unidade curricular, onde o foco passava por aspetos de disponibilidade e/ou desempenho. Os passos considerados para alcançar o objetivo do trabalho, realizados num contexto virtualizado, foram os seguintes:

- Preparação e configuração do ambiente de virtualização (*Proxmox*) a utilizar para a criação dos *clusters*.
- Implementar em cada *Proxmox node* armazenamento compartilhado.
- Realizar testes experimentais para testar a capacidade dos *clusters* em manter a sua disponibilidade.

1.3 Estrutura do Relatório

Este relatório encontra-se organizado nos seguintes seis capítulos:

- *Capítulo 1* - Neste capítulo é feita uma introdução à envolvente geral do projeto e objetivos do mesmo. É ainda apresentada a estrutura do documento.
- *Capítulo 2* - Neste capítulo são apresentados conceitos essenciais para uma melhor compreensão do trabalho desenvolvido, essenciais para que seja possível entender alguns conteúdos abordados na parte experimental do projeto.
- *Capítulo 3* - Neste capítulo é feito um levantamento dos requisitos funcionais e das ferramentas a utilizar que são capazes de dar resposta aos vários objetivos do projeto.
- *Capítulo 4* - Neste capítulo é feita uma exposição da topologia desenvolvida para efeitos de teste das ferramentas que se pretendem estudar. São ainda indicadas as experiências a serem alvo de teste em termos práticos.
- *Capítulo 5* - Neste capítulo encontra-se o guião laboratorial elaborado com as etapas seguidas, juntamente com auxiliares visuais, para a realização com sucesso dos testes propostos.
- *Capítulo 6* - Neste capítulo são descritos os objetivos alcançados e é feita uma retrospectiva acerca do trabalho desenvolvido. São ainda apresentados alguns pontos a endereçar em trabalho futuro.

Para além destes capítulos existem ainda as *Referências*, que se encontram no final do documento, onde é possível encontrar todas as fontes consultadas e das quais foram retiradas informação para a redação do relatório.

Capítulo 2

Conceitos

Neste capítulo serão apresentados alguns conceitos e terminologias essenciais para a compreensão do projeto desenvolvido.

2.1 *Cluster*

São colecções de computadores ou servidores autónomos que estão ligados em rede. Estas máquinas interligados estão equipadas com software para coordenar programas nesses computadores (ou entre eles) e podem, por conseguinte, trabalhar em conjunto para realizar tarefas computacionalmente intensivas. [2]

2.2 *High Availability*

Também denominado de *fault tolerance*, no contexto do presente projeto, este princípio consiste em servidores agrupados para funcionarem como um sistema único e unificado.

A utilização deste mecanismo de elevada disponibilidade ajuda a garantir que não existe um ponto único de falha para as infraestruturas críticas e reduz ou elimina o tempo de inatividade. Isso significa que, se um servidor no *cluster* falhar, outro servidor ou nó pode assumir o controlo imediatamente para ajudar a garantir que a aplicação ou serviço suportado pelo *cluster* permanece operacional. [1]

2.3 *Nested Virtualization*

Técnica que permite executar uma VM dentro de outra VM, que se encontra a correr num *host* físico. Isto significa que não é necessário desinstalar o sistema operativo do *host* e instalar outro desejado ou mesmo recorrer ao *dual boot*. Este método revela-se útil no contexto deste projeto pois permite testar e configurar software num ambiente isolado

sem perigo de afetar o *host*, para além de permitir isolar uma VM de outra para evitar que um sistema operativo corrompa outro.

2.4 *Quorum disk*

De forma a perceber o conceito de *quorum disk* é necessário primeiro perceber o conceito de *quorum*, que é um mecanismo que determina qual o nó que deve controlar os recursos do *cluster* no evento de uma falha.

Para esse tipo de ocorrências existe este tipo de disco especial, usado numa situação de *failover* num *cluster*, de modo a assegurar que o *cluster* consegue manter o *quorum* impedindo que haja corrupção de dados ou inconsistências quando os nós não se conseguem comunicar. [4]

Capítulo 3

Tecnologias Usadas

Para ser possível realizar o conjunto de experiências que se sucedem nos seguintes capítulos foi necessária uma instalação prévia, de um conjunto de serviços e aplicações, com ferramentas a seu dispor que permitam explorar a implementação de *clusters*. De modo a dar os fundamentos necessários para ser possível realizar com sucesso esta fase independente do projeto, foram utilizadas as seguintes ferramentas.

3.1 *Proxmox*

Entre os vários candidatos para escolha de sistema virtualizador de tipo um, o eleito foi o *Proxmox VE*, baseado numa arquitetura em *Debian/Linux*, pois é um software *Open Source* bastante completo que permite criar e gerir VM e *containers* num ou em múltiplos *hosts*. Para além do anterior referido, este oferece funcionalidades para gestão de *clusters* como armazenamento compartilhado, *backups* entre outras, que se podem revelar úteis no desenvolvimento do projeto.[8]

3.2 *High-Availability Cluster*

É um conjunto de servidores, ou outro tipo de máquinas que trabalham em conjunto e que partilham recursos funcionando como um único sistema para fornecer um serviço ou aplicação com alta disponibilidade. A utilização de um sistema deste tipo permite manter o serviço ou aplicação sempre disponível, mesmo em caso de falha de um ou mais componentes do *cluster*, garantindo assim que há sempre a possibilidade de aceder ao serviço através de uma das máquinas.[6]

3.3 *Ceph*

O *ceph* fornece armazenamento de objetos distribuídos e sistemas de ficheiros em larga escala, de modo a fornecer um alto desempenho. A integração do mesmo no *Proxmox* oferece uma solução poderosa no que toca ao armazenamento, pois permite que os dados sejam replicados entre vários nós e facilita a migração de VM.[9]

Capítulo 4

Planeamento

Ao longo deste capítulo serão apresentadas a infraestrutura planeada para teste, bem como os seus constituintes e as propostas de experiências a realizar no capítulo seguinte.

4.1 Ambiente laboratorial

Para possibilitar a avaliação da do ambiente desenvolvido é essencial, numa primeira fase, perceber quais os componentes do mesmo e como estes se interligam. A Figura 4.1 ilustra o esquema de funcionamento de como este se encontra composto para realização de testes.

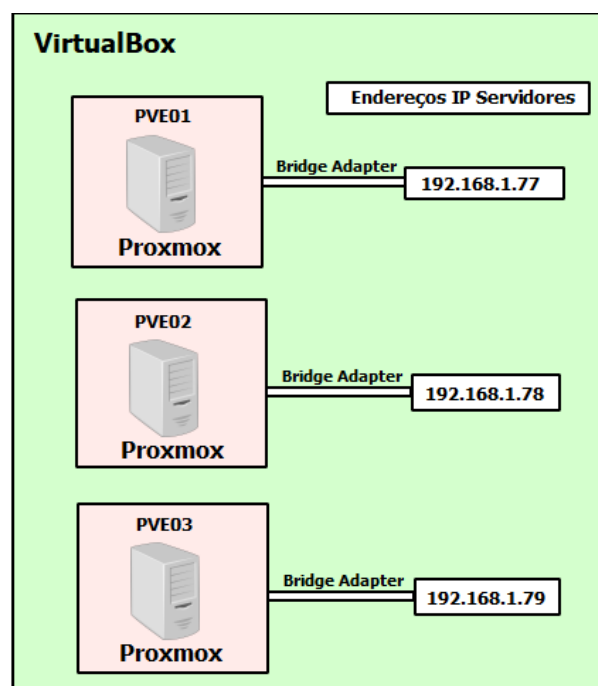


Figura 4.1: Topologia experimental

4.2 Experiências

Alguns dos testes que se pretendem realizar, no *cluster* criado, para compreensão do funcionamento interno do *Proxmox* e quais as suas limitações, bem como os seus pontos fortes:

- ***Cluster Creation*** - Criação de um *cluster* num dos servidores e *join* dos restantes nós a esse mesmo *cluster*, com base na informação copiada do primeiro (*copy information*).
- ***Node Failure*** - Desligar um dos nós do *cluster* manualmente e observar como ele reage à falha do nó, como ele redistribui as máquinas virtuais e os recursos para outros nós. Por fim, verificamos se os dados estão acessíveis após a falha do nó e se o armazenamento compartilhado está a ser usado de forma eficaz.
- ***Live migration*** - Fazer manualmente uma migração de um dos *clusters* para o outro. Após a migração, verificar se a VM está a funcionar corretamente no novo nó e se há conectividade na rede.

Capítulo 5

Guião Laboratorial

Neste capítulo será documentado todo o processo de configuração e testes feitos na topologia anteriormente apresentada de modo a testar a sua resiliência e capacidade de recuperação de *clusters* em cenários de falhas simuladas. As experiências incluídas ajudarão a avaliar como estes conseguem lidar com falhas de nós individuais, falhas de rede, sobrecarga de recursos e migração ao vivo. No fim de cada experiência serão apresentados os resultados (com auxiliares gráficos) e a conclusão retirada no final de cada teste.

5.1 Instalações necessárias

Para ser possível realizar o conjunto de experiências que se sucedem nas seguintes secções foi necessária uma instalação prévia, de um conjunto de serviços e aplicações, com ferramentas a seu dispor que permitam explorar a criação e gestão do *cluster*.

5.2 VirtualBox

De modo a não ter de recorrer a *dual boot* para instalação do ambiente *Proxmox*, devido a riscos de alguma configuração incorreta comprometer o sistema hospedeiro, foi optado por usar, o mecanismo de *Nested Virtualization* anteriormente referido. Para aplicar essa técnica foi utilizado como software de virtualização o *VirtualBox*, que permitiu ter acesso a várias funcionalidades úteis no decorrer de qualquer experimentação pela primeira vez, como *Snapshots* e a criação de duas interfaces de rede para comunicação entre servidores:

A ter em conta aquando da instalação desta ferramenta foi a versão a usar, um fator bastante crucial, dado que esta tem de ter nas suas funcionalidades a opção de permitir ativar numa VM a *Nested virtualization*, conceito abordado no capítulo 2.

5.3 Proxmox

Apesar de haver uma versão mais recente do que a utilizada para o projeto, a escolha do ficheiro ISO para a instalação do *Proxmox* foi a *version 7.4-1*, atualizada em março de 2023. Tal escolha passou pelo facto de a documentação ser mais fiável e não haver perigo de algum tipo de erro que possa ter havido por parte dos desenvolvedores, algo comum em versões recentemente lançadas onde pode não haver tanta estabilidade.

Para proceder à instalação propriamente dita serão fornecidos os passos, com auxílio visual de imagens, que foram percorridos no decorrer do desenvolvimento do projeto, tendo em conta que o mesmo processo pode ser feito através da linha de comandos do *Proxmox* sem qualquer *Graphical User Interface* (GUI). Uma fonte de informação importante seguida nesta fase do projeto, passou pela documentação oficial da tecnologia que pode ser encontrada no *link* [7]

5.3.1 Criação da *Virtual Machine*

Ao seleccionar a opção *new* no *VirtualBox*, foi aberta uma janela separada onde são feitas as primeiras configurações e fornecida a ISO a ser usada. O sistema operativo seleccionado teve em consideração que o *Proxmox* é baseado em *Debian 11*. Tais configurações encontram-se ilustradas na Figura 5.1.

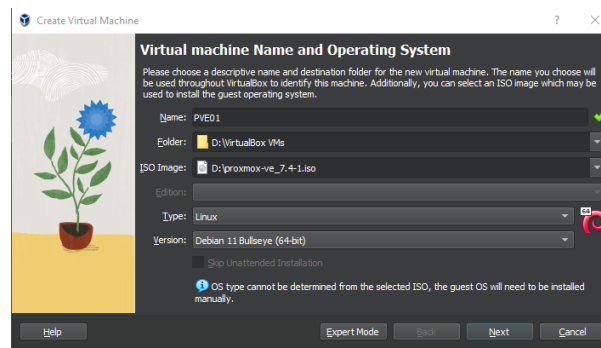


Figura 5.1: Instalação VM *Proxmox*

As definições finais, após o ajuste de alguns parâmetros adicionais, como o espaço em disco, a memória *Random Access Memory* (RAM) e os núcleos do processador foram as que mostra a Figura 5.2.

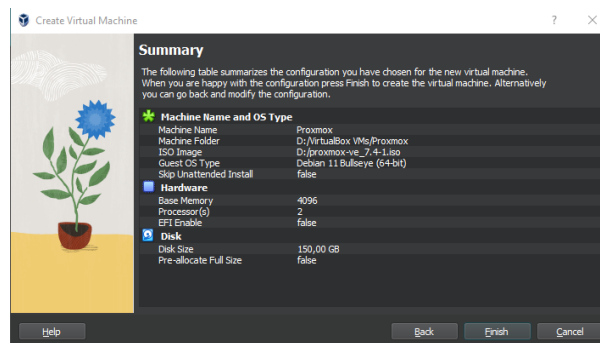


Figura 5.2: Configurações finais da VM

Ainda na fase de instalação inicial, foi necessário nas definições pós instalação da VM, como é representado na Figura 5.3, manualmente confirmar que a máquina deve disponibilizar *Nested Virtualization*, pelos motivos explicados no capítulo 2.

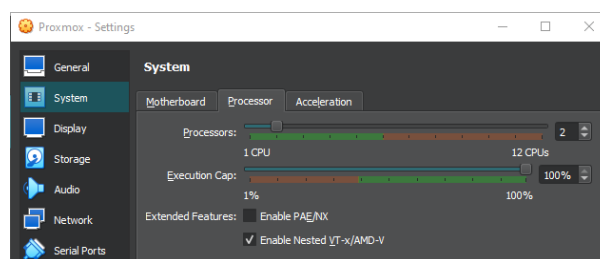


Figura 5.3: Ativação da *Nested Virtualization*

Por último, nas definições de rede, a placa de rede que se encontrava ativa por omissão era uma placa *Network Address Translation* (NAT), o que não permitia aceder à VM do *Proxmox* diretamente através do *browser*. Para possibilitar tal funcionalidade, apenas houve a alteração para uma placa *Bridge Adapter*, como mostra a Figura 5.4.

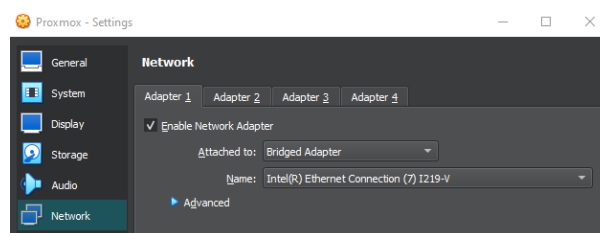


Figura 5.4: Configuração da placa de rede da VM

5.3.2 Instalação do *Proxmox*

Ao inicializar a VM foi pedido para inicializar a instalação e, nesse processo ilustrado na Figura 5.5, é possível reparar no *Proxmox* a fazer um pedido DHCP de forma a descobrir um servidor capaz de lhe fornecer um endereço IP.

```
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 3
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 7
DHCPOFFER of 192.168.1.221 from 192.168.1.1
DHCPREQUEST for 192.168.1.221 on enp0s3 to 255.255.255.255 port 67
DHCPACK of 192.168.1.221 from 192.168.1.1
```

Figura 5.5: Obtenção de um IP através de DHCP

Após esse passo, apenas foi necessário avançar visto que o disco rígido correto, com o tamanho definido na criação da VM, já se encontrava selecionado por omissão como pode ser visto na Figura 5.6.



Figura 5.6: Seleção do disco rígido

Foi selecionado o país de residência, como ilustra a Figura 5.7 de modo a acertar fuso horário que poderá afetar serviços como o *Network Time Protocol* (NTP), que é um fator crítico quando se pretende implementar *File Sharing*.

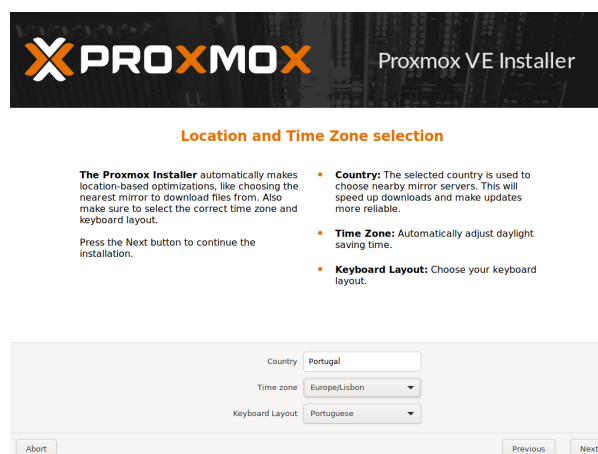


Figura 5.7: Configuração do fuso horário e localização

Na definição da palavra-passe não foi necessário indicar um nome de utilizador, como mostra a Figura 5.8 pois numa instalação *Proxmox* já se encontra associado um utilizador por omissão, o utilizador *root*, sendo as credenciais de acesso a *logins* que sejam necessários ambos os parâmetros anteriormente referidos.

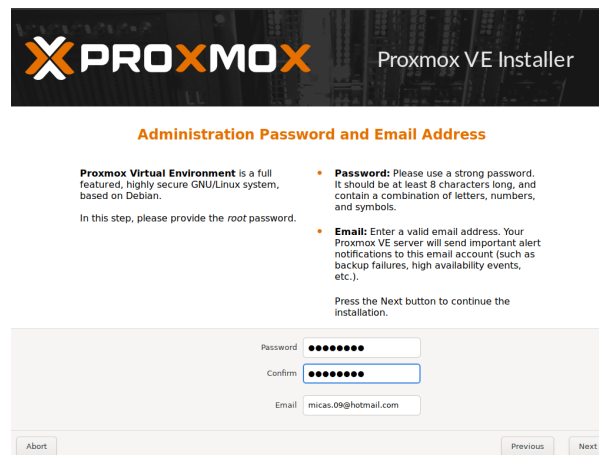


Figura 5.8: Definição de *password* e *email*

Neste passo, como a atribuição de um endereço IP por parte do servidor *DHCP* foi bem-sucedida não houve a necessidade de configuração manual, apenas houve a alteração do *hostname*, como se encontra ilustrado na Figura 5.9.

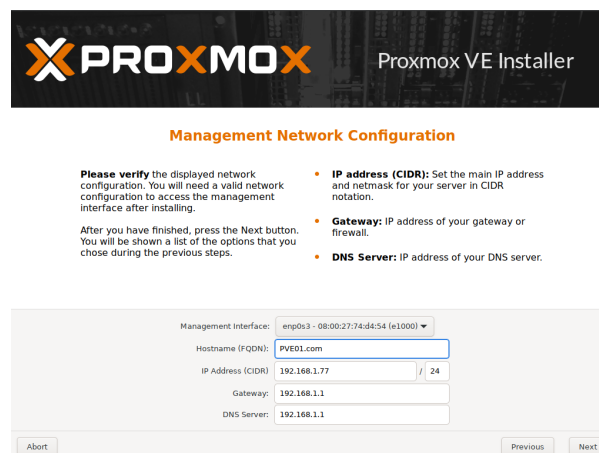


Figura 5.9: Configurações da rede

O servidor, quando dadas por concluídas todas as alterações, procedeu a reiniciar automaticamente o que levou à apresentação do ecrã de instalação inicial novamente. Para ser possível aceder ao *Proxmox*, propriamente dito, e começar a configurar o *cluster* foi necessário remover o ficheiro ISO do armazenamento da VM, como pode ser visto na Figura 5.10.

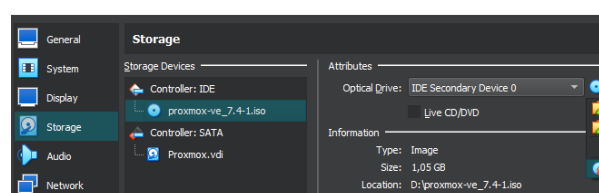


Figura 5.10: Remoção do ficheiro ISO

Uma vez que o objetivo do trabalho passava pela criação de um *cluster* de várias

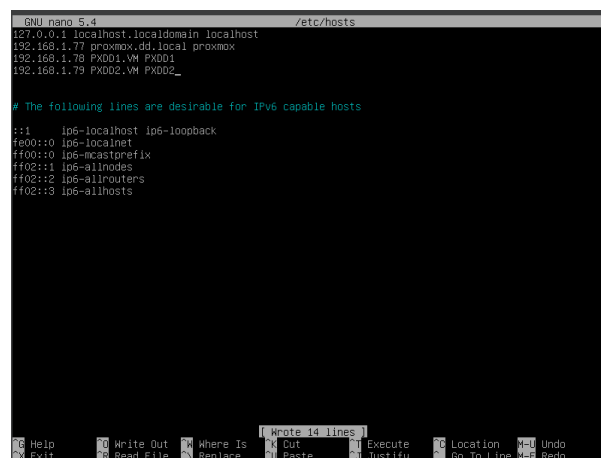
máquinas em *Proxmox*, serão precisos pelo menos três servidores a correr em simultâneo. Este número deve-se ao facto de que para um *cluster* com apenas dois nós, seria necessário haver a configuração de um *quorum disk*, conceito explicado em capítulos anteriores, o que dificultaria a experimentação laboratorial de cenários de falhas e migração.

Com isto em mente, foram criadas mais duas máquinas virtuais com o *Proxmox*, repetindo todos os passos anteriores para cada uma, levando a uma topologia final semelhante à representada na Figura 4.1.

5.4 Configuração do ficheiro *hosts*

Embora seja comum referenciar todos os nomes dos nós e os respetivos endereços IP no ficheiro `/etc/hosts`, este passo não é obrigatório para o funcionamento do *cluster*, apesar de ser recomendado caso seja necessário efetuar uma conexão via *Secure Shell* (SSH) de um nó para outro.

Este ficheiro, no *Proxmox*, assim como em outros sistemas *Linux* e/ou *Unix*, é utilizado para mapear endereços IP para nomes de *host*. Em todos os três nós são adicionadas as informações acima mencionadas de ambos os outros servidores, levando a um ficheiro como o ilustrado na Figura 5.11.



```
GNU nano 5.4 /etc/hosts
127.0.0.1 localhost.localdomain localhost
192.168.1.77 proxmox.dd.local proxmox
192.168.1.78 PX001.VM PX001_
192.168.1.79 PX002.VM PX002_

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe80:: ip6-localhost
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Figura 5.11: Adicionar servidores ao ficheiro *hosts*

5.5 Criação de um *Cluster*

Um *Cluster* no *Proxmox* refere-se à configuração de vários nós (*hosts*) unidos para formar um ambiente de computação distribuída. Ao criar um *Proxmox Cluster*, é possível aproveitar recursos como alta disponibilidade e migração de máquinas virtuais entre os nós.

De entre os três servidores disponíveis foi preciso fazer a escolha de um deles para funcionar como o *master node*, que ficaria responsável por manter o estado do *cluster*

nele criado. Por outro lado, os restantes nós ficaram a funcionar como *worker nodes*, que desempenhavam as tarefas atribuídas pelo *master* e que não possuem um *cluster* próprio mas, que se juntam ao *cluster* criado no servidor principal.

Por motivos de conveniência, a seleção passou pela primeira máquina *Proxmox* instalada, embora a escolher pudesse passar por qualquer uma das restantes. O processo de criação do *Cluster* é o demonstrado na Figura 5.12, sendo que no final deste passo é fornecido um aviso de "*TASK OK*" indicando o sucesso da operação.

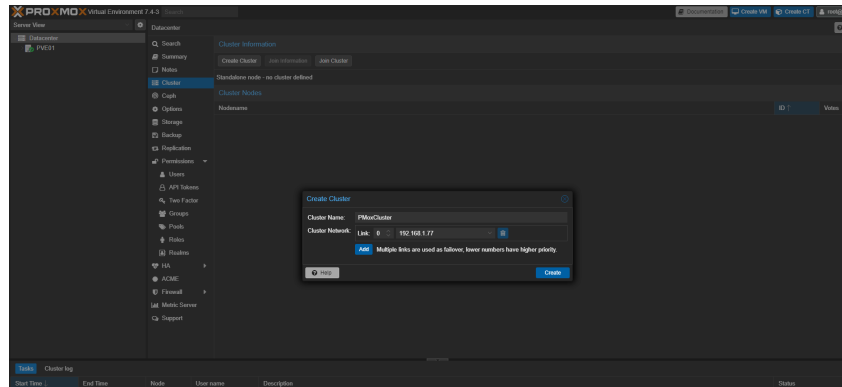


Figura 5.12: Criação do *Cluster*

Após a criação bem sucedida do *Cluster*, de maneira a que os outros nós tenham acesso a ele, foi copiada a informação do mesmo que é apenas uma chave *Application Programming Interface* (API), como mostra a Figura 5.13, obtida ao clicar no botão *join information*.

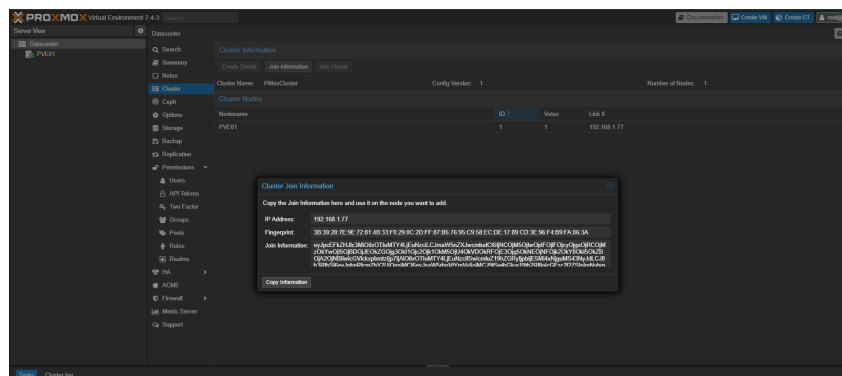


Figura 5.13: Cópia da informação do *cluster*

O processo seguinte, de associação de ambas as máquinas restantes ao *cluster* criado, foi feito da mesma maneira para ambos os servidores. Nesta etapa, como pode ser visto na Figura 5.14, foi apenas necessário inserir a chave copiada anteriormente e colocar a palavra-passe do cliente *root* do nó definido como *master*.

Após a junção do nó ao *cluster*, o seu certificado atual é substituído por um atribuído pela *Certificate Authority* (CA) do *cluster*. Isto significa que, apesar de aparecer uma notificação de "*Join request OK*", por vezes sucede uma falha momentânea de conectividade

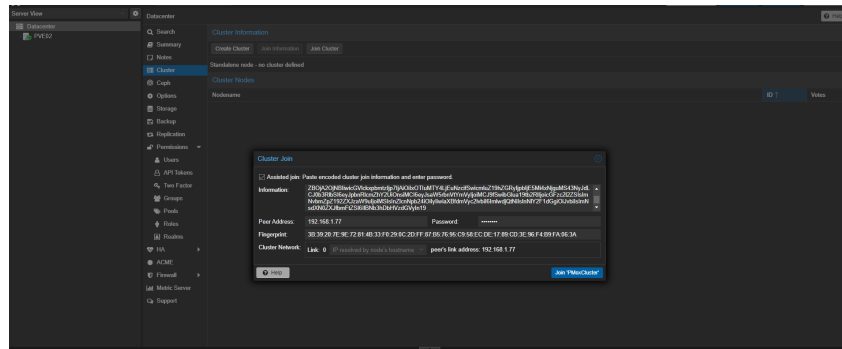


Figura 5.14: Adesão ao *Cluster*

entre os nós. Para resolução deste problema basta proceder à atualização da interface *web* correspondente ao novo nó que se juntou ao *cluster*.

Embora a criação do *cluster* feita no servidor um, e que os restantes se tenham juntado a este, é possível a gestão de toda a infraestrutura a partir da interface *web* de qualquer uma das máquinas. Na Figura 5.15 é observável um exemplo disso mesmo, onde é mostrado o segundo nó criado que possui todas as restantes na sua interface.

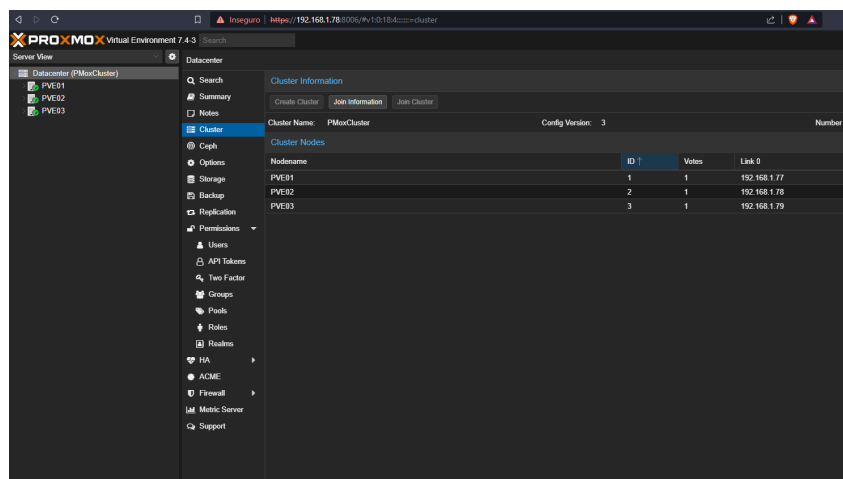


Figura 5.15: Interface do nó *PVE02*

5.6 Gestão de utilizadores

Este é um aspeto crucial no que toca à segurança do ambiente de virtualização. Com a configuração de vários utilizadores é possível definir funções, atribuir permissões e restringir o acesso a recursos específicos, assegurando que apenas indivíduos autorizados podem executar acções que possam afetar a estabilidade ou a segurança da topologia.

Para efeitos de experimentação foram criados dois utilizadores, um *Proxmox user* e um *PAM user* em que ambos conseguem aceder ao sistema *Proxmox*, a única diferença passa pelo local onde o utilizador é criado e onde essa mesma informação é guardada.

Os utilizadores PAM são criados no sistema *Linux* e adicionados ao *Proxmox* como um utilizador PAM, enquanto os utilizadores *Proxmox* são criados no próprio *Proxmox*. Os utilizadores criados no domínio do *Proxmox* não têm um utilizador de sistema correspondente no servidor *Linux* subjacente, o que torna os utilizadores PAM mais adequados para quem acede ao servidor através de SSH.

5.6.1 Criação do utilizador *PAM*

Para criar um utilizador bastou, com o *Datacenter* do *cluster* selecionado, ir à aba dos *users* e proceder à introdução das informações, como mostra a figura 5.16.

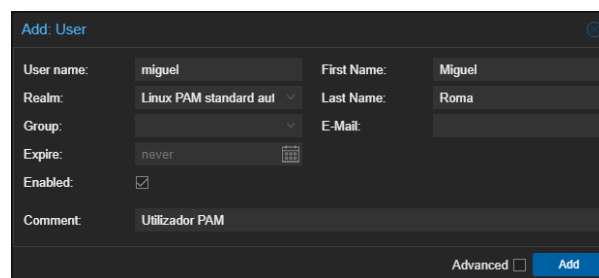
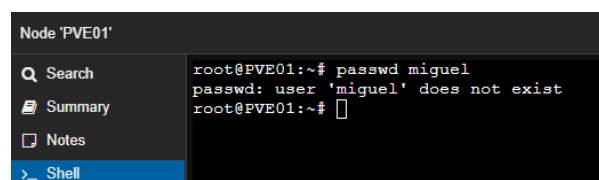


Figura 5.16: Criação utilizador PAM

Algo importante, que foi possível reparar ao verificar o ficheiro */etc/passwd*, responsável por guardar contas do tipo *Linux* como é o caso de uma conta PAM, foi o facto de o utilizador criado não ser guardado no final do ficheiro, que seria o sítio habitual seguindo os padrões. Isso significa que, apesar de o utilizador ter sido criado através do GUI não foi criado no sistema, como pode ser verificado na Figura 5.17.



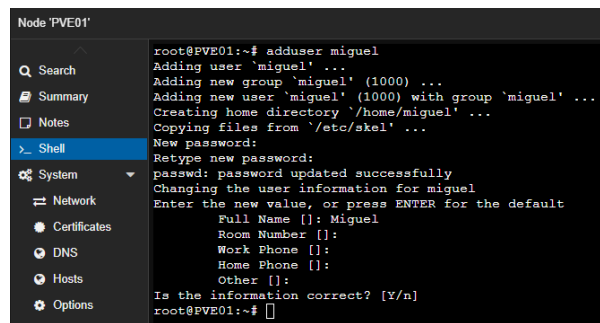
```
Node 'PVE01'
Search
Summary
Notes
Shell

root@PVE01:~# passwd miguel
passwd: user 'miguel' does not exist
root@PVE01:~#
```

Figura 5.17: Utilizador PAM não existe

Tentativas de início de sessão com este utilizador não vão ser válidas porque para o servidor o utilizador não existe. Para resolução deste "erro", é necessário recorrer à *shell* e, com a introdução manual de certos comandos ilustrados na Figura 5.18, o utilizador fica assim efetivamente criado.

Este tipo de utilizadores são, normalmente, utilizados para acesso ao sistema *Proxmox*, através da utilização de SSH.



```

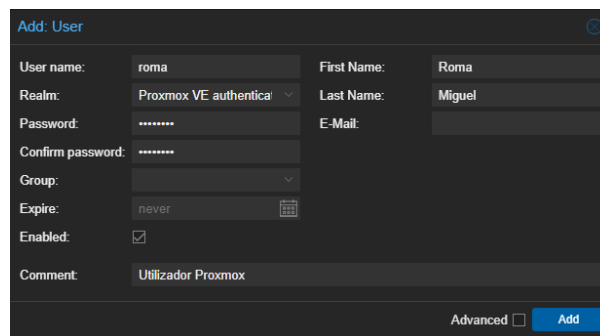
Node 'PVE01'
root@PVE01:~# adduser miguel
Adding user 'miguel' ...
Adding new group 'miguel' (1000) ...
Adding new user 'miguel' (1000) with group 'miguel' ...
Creating home directory '/home/miguel' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for miguel
Enter the new value, or press ENTER for the default
Full Name []: Miguel
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n]
root@PVE01:~#

```

Figura 5.18: Criação do utilizador PAM através da *shell*

5.6.2 Criação do utilizador *Proxmox*

O processo de criação deste tipo de utilizadores foi bastante mais simples, sendo apenas necessária a introdução das credenciais de acesso como mostra a Figura 5.19, sem ter de recorrer a métodos de comandos.



The 'Add User' form in Proxmox VE includes the following fields and values:

- User name: roma
- First Name: Roma
- Realm: Proxmox VE authenticat...
- Last Name: Miguel
- Password: [masked]
- E-Mail: [empty]
- Confirm password: [masked]
- Group: [empty]
- Expire: never
- Enabled: ☒
- Comment: Utilizador Proxmox

Buttons at the bottom: Advanced ☐ Add

Figura 5.19: Criação do utilizador *Proxmox*

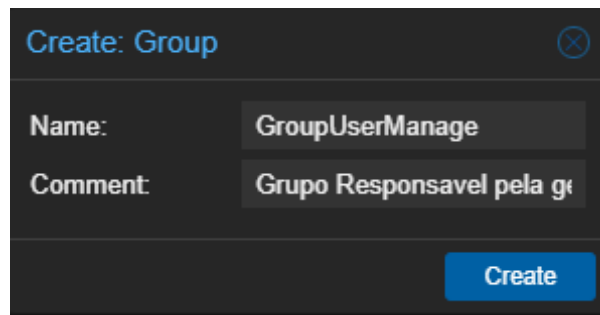
Este tipo de utilizadores são, normalmente, utilizados para gestão do sistema *Proxmox*. Numa situação hipotética onde se pretende, como gestor do ambiente, convidar um colega a fazer gestão das VM através da interface *web* é este método que tem de ser utilizado.

Após conclusão dos passos acima, os utilizadores ficaram aptos a serem utilizados e foi possível o início de sessão no servidor onde estes foram criados, neste caso o servidor *PVE01*, apesar de não terem ainda qualquer tipo de permissões associadas.

5.6.3 Criação de um grupo

A criação de um grupo é essencial no que toca à gestão de utilizadores, pois, por vezes, não se querem fornecer permissões de alto nível a certos utilizadores mas existem outros que precisam dessas permissões por terem cargos mais elevados ou que precisam delas para gerir o sistema. Por esse motivo, foi necessário criar um grupo, onde pudessem ser incluídos apenas os utilizadores pretendidos, de modo a separá-los de outros, como mostra a Figura 5.20.

Para efeitos de teste, apenas foi concedido a este grupo a possibilidade de gerir os



Create: Group

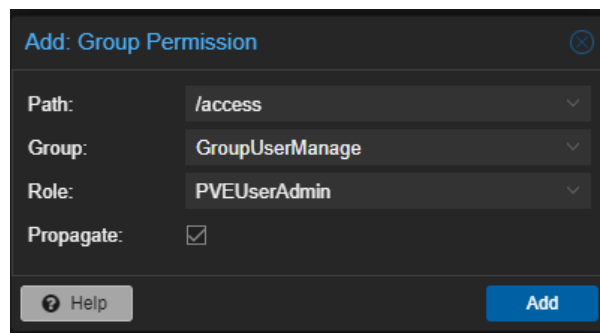
Name: GroupUserManage

Comment: Grupo Responsavel pela g

Create

Figura 5.20: Criação de um grupo

utilizadores novos criados, sendo que não vão ter acesso nem a qualquer VM nem outro tipo de dados como o armazenamento, entre outros. Essa configuração pode ser vista na Figura 5.21.



Add: Group Permission

Path: /access

Group: GroupUserManage

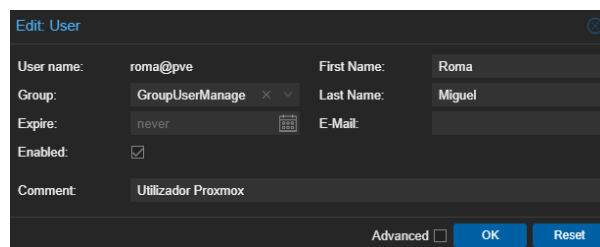
Role: PVEUserAdmin

Propagate: ☒

Help Add

Figura 5.21: Permissões do grupo

Por fim, foi adicionado o utilizador desejado a esse grupo para que ele possa usufruir das novas permissões, como mostra a Figura 5.22.



Edit: User

User name: roma@pve First Name: Roma

Group: GroupUserManage Last Name: Miguel

Expire: never E-Mail:

Enabled: ☒

Comment: Utilizador Proxmox

Advanced ☐ OK Reset

Figura 5.22: Permissões do utilizador

É possível confirmar que a experiência foi concluída com sucesso quando ao iniciar sessão com o utilizador este apenas consegue ver a informação que lhe é permitida. O ecrã visível por este é o demonstrado na Figura 5.23.

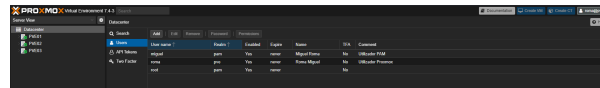


Figura 5.23: Vista do perfil do utilizador

5.7 Rede separada do *Cluster*

Em servidores *Proxmox*, é altamente recomendado o uso de no mínimo duas redes que funcionem independentemente uma da outra, uma vez que traz diversos benefícios a nível de redundância e alta disponibilidade. Para além desse fator, se uma placa de rede falhar, a outra pode assumir a carga de tráfego, mantendo a conectividade. Na figura 5.24 e 5.25, é possível ver a segunda placa a ser adicionada, bem como a sua configuração no ficheiro *interfaces*.

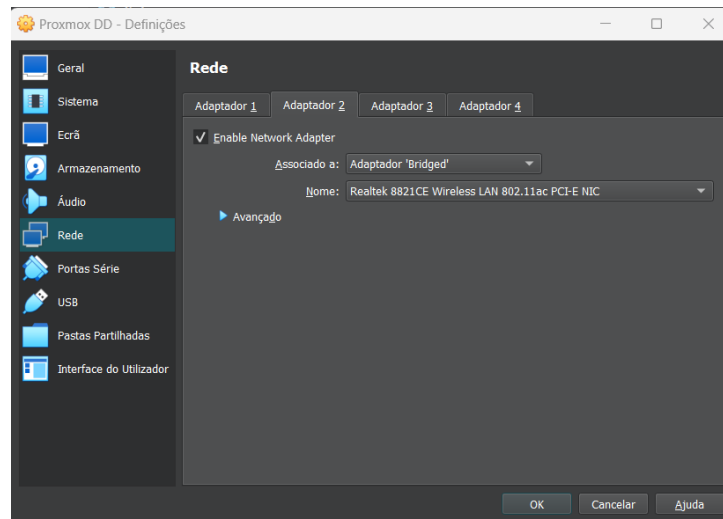


Figura 5.24: Adicionar a segunda placa aos servidores

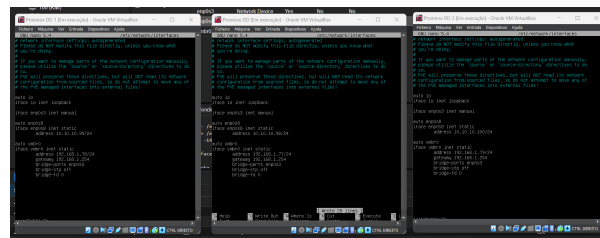


Figura 5.25: Configuração da segunda interface

Na figura 5.26, é possível ver a conectividade entre os três servidores.

```
root@proxmox:~# ping 192.168.1.78
PING 192.168.1.78 (192.168.1.78) 56(84) bytes of data.
64 bytes from 192.168.1.78: icmp_seq=1 ttl=64 time=0.708 ms
^C
--- 192.168.1.78 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.708/0.708/0.708/0.000 ms
root@proxmox:~# ping 192.168.1.79
PING 192.168.1.79 (192.168.1.79) 56(84) bytes of data.
64 bytes from 192.168.1.79: icmp_seq=1 ttl=64 time=0.454 ms
64 bytes from 192.168.1.79: icmp_seq=2 ttl=64 time=0.229 ms
^C
--- 192.168.1.79 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1016ms
rtt min/avg/max/mdev = 0.229/0.341/0.454/0.112 ms
```

Figura 5.26: Conectividade entre os servidores

5.8 Criação de uma máquina Virtual

Para cumprimento de uma das experimentações propostas, a migração entre servidores, é necessário pelo menos uma VM instalada numa das máquinas do *cluster*. Em primeiro lugar, precisamos de dar *upload* ao ISO do *Kali Linux* (máquina escolhida para o efeito), como mostra na 5.27. Só de seguida, é que passamos para a parte de configuração do mesmo, como mostram as figuras 5.28 e 5.29.

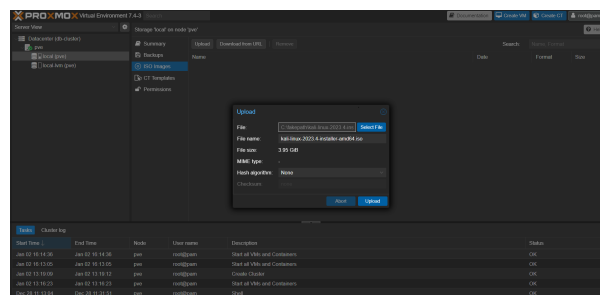


Figura 5.27: Seleção da ISO a usar

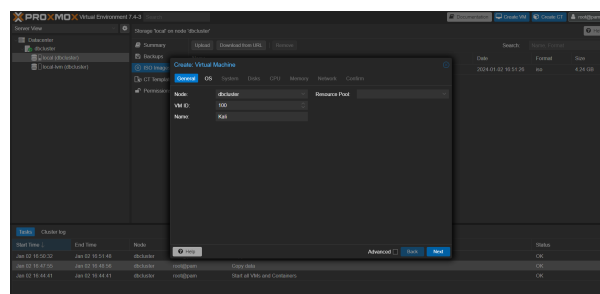


Figura 5.28: Criação de uma Máquina Virtual

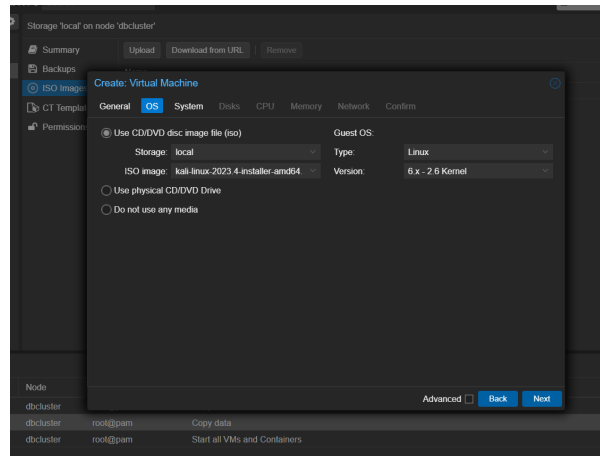


Figura 5.29: Criação de uma Máquina Virtual

Após o *upload* do ficheiro ISO, passamos para a configuração do *Kali Linux* no *prox-mox node*. A figura 5.30 apresenta os diferentes modos de configuração.

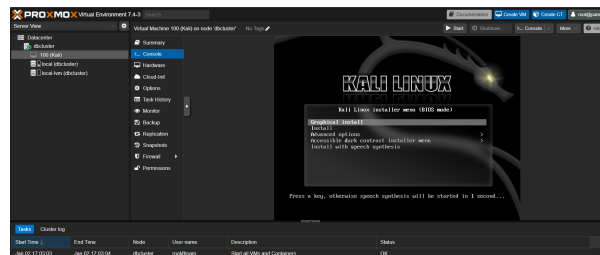


Figura 5.30: Configuração do Kali

Depois de concluídos os passos acima, temos assim a máquina instalada com sucesso, como podemos ver na figura 5.31.

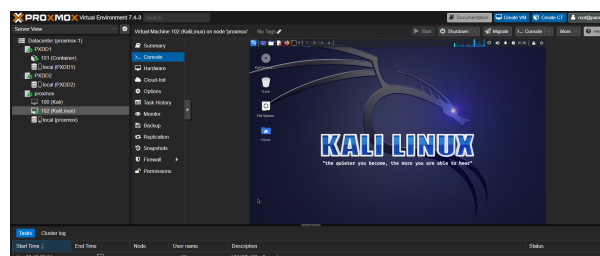


Figura 5.31: *Kali Linux*

5.9 Ceph

Para proporcionar o fator de HA é necessária a instalação e configuração do *ceph*. O *Ceph* é frequentemente utilizado em ambientes que exigem alta disponibilidade de armazenamento. Isto significa que os dados são distribuídos entre vários nós num *cluster*

e, mesmo na eventualidade de um nó falhar, os dados irão continuar disponíveis noutros nós, garantindo que a operação continue sem interrupções. [3] [5] [10]

5.9.1 Instalação nos nós

Um pormenor tido em conta nesta fase foi assegurar que, a versão selecionada para instalação do *Ceph* era a *Ceph Quincy* como demonstrado na Figura 5.32. Esta versão não só é a mais recente, como também a mais estável e com uma melhoria de performance comparativamente a outras versões. O processo a seguir descrito foi efetuado em todos os três nós, uma vez que o objetivo é proporcionar o fator de disponibilidade.

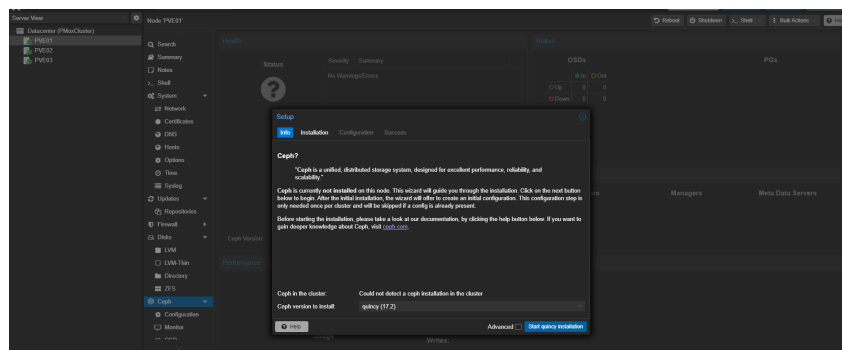


Figura 5.32: Escolha de instalação do *Ceph*

5.9.2 Criação do *OSD*

Para configurar esta funcionalidade foi necessário, em todos os servidores, criar um novo disco destinado apenas a este propósito e, no processo de instalação do OSD, selecionar a nova *drive* que se pretende usar como mostra a Figura 5.33.

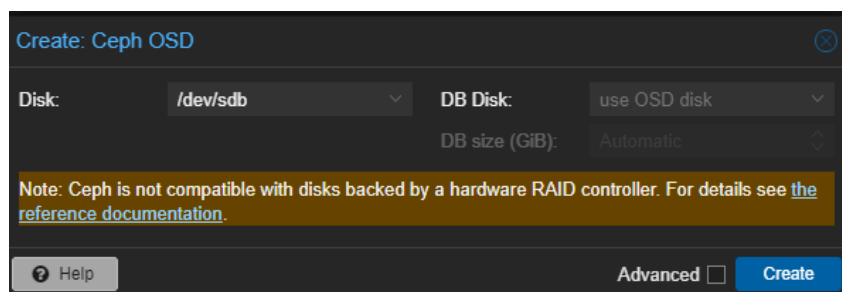


Figura 5.33: Escolha do novo disco criado

Feita a criação em cada nó do *cluster*, a configuração final foi a que mostra a Figura 5.34, sendo que o nó que dispõe a seguinte informação é o *PVE03*.

Name	Class	OSD Type	Status	Version	weight	rowweight	Used (%)	Total	Apply/Commit Latency (ms)	PGs
default										
PVEB3	hdd	Blockstore	down	17.2.6	0.0488	1.00	0.00	0 B	0 / 0	0
PVEB2	hdd	Blockstore	up	17.2.6	0.0488	1.00	0.04	50.00 GB	0 / 0	0
osd.1	hdd	Blockstore	up	17.2.6	0.0488	1.00	0.04	50.00 GB	0 / 0	0
PVEB1	hdd	Blockstore	up	17.2.6	0.0488	1.00	0.04	50.00 GB	0 / 0	0
osd.0	hdd	Blockstore	up	17.2.6	0.0488	1.00	0.04	50.00 GB	0 / 0	0

Figura 5.34: Configuração OSD final

5.9.3 Criação de uma *Pool*

A funcionalidade da *pool* será proporcionar um armazenamento compartilhado que os três nós possam usar para guardar os discos das VM. A criação foi bastante simples sendo apenas necessária a informação do nome da *pool*, como mostra a Figura 5.35

Create: Ceph Pool

Name: PG Autoscale Mode:

Size: Add as Storage: ☒

Figura 5.35: Criação da *Ceph pool*

5.9.4 Configuração do monitor

Por omissão a monitorização do *ceph* só se encontra configurada para o primeiro servidor onde este foi instalado, neste caso o nó *PVE01*. Isto pode levar a problemas de disponibilidade no caso de falha deste mesmo nó, levando a que o *Ceph* não consiga fazer ações de sincronização, produzindo dados errados e desconfigurados.

Para que não surjam este tipo de problemas, foi preciso configurar em cada nó o seu próprio monitor de acordo com a Figura 5.36.

Name	Host	Status	Address	Version	Quorum
mon.PVE01	PVE01	running	192.168.1.77:6789	17.2.6	Yes
mon.PVE02	PVE02	running	192.168.1.78:6789	17.2.6	Yes
mon.PVE03	PVE03	running	192.168.1.79:6789	17.2.6	Yes

Figura 5.36: Monitores criados no *Ceph*

5.10 Criação de um *Container*

Um *container* fornece virtualização ao nível do sistema operativo. Para criar um *container*, em primeiro lugar, teve de ser instalado um *template*, que posteriormente vai ser adicionado ao *container*, aquando da sua criação. Os *templates* não são mais do que imagens pré-configuradas de sistemas operativos que podem ser usadas para criar

rapidamente novos *containers*. Têm a vantagem de ajudar a simplificar o processo de implementação, uma vez que não é preciso começar do zero para configurar um ambiente.

No caso do presente projeto, foi escolhido o *ubuntu-22.04-standard*, como sistema que a usar no *container*. Tal pode ser visto na Figura 5.37.

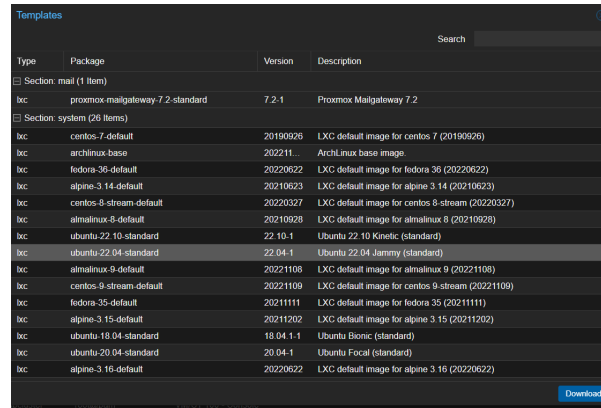


Figura 5.37: Instalação de um *template*

De seguida, procedemos à instalação do *container* e associamos-lhe o *template*, tal como mostram as figuras 5.38 e 5.39.

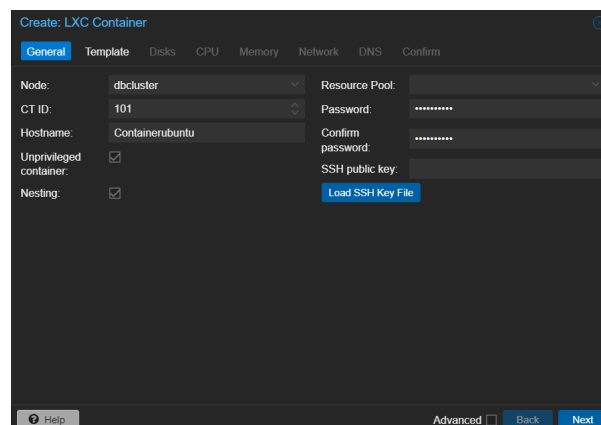


Figura 5.38: Criação de um *container*

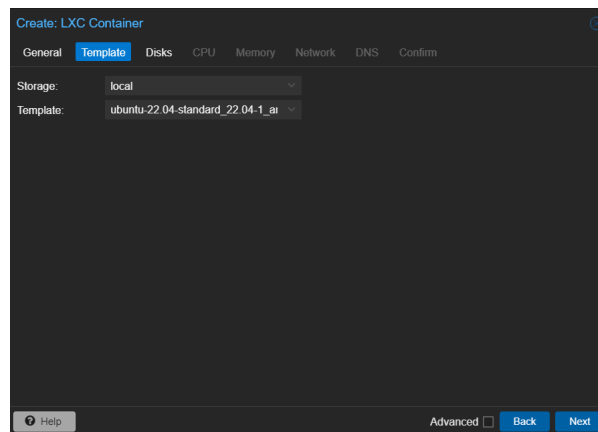


Figura 5.39: Associação do *template* ao *container*

Depois de criado o *container*, foi apenas necessário fazer o *login* com as credenciais escolhidas para ter acesso à *shell* do mesmo, sem qualquer configuração adicional do ambiente, uma vez que a imagem já vinha pré-configurada, como foi referido. É possível ver a *shell* do *container* através da figura 5.40.

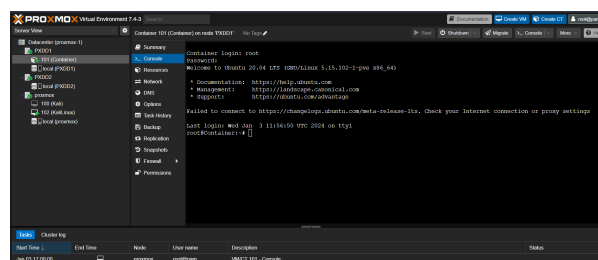


Figura 5.40: *Shell* do *container*

5.11 *High-Availability group*

No contexto do *Proxmox*, HA refere-se à capacidade de fornecer alta disponibilidade para VMs e *containers*, garantindo que continuem operacionais, mesmo em caso de falha de *hardware* ou *software*.

Grupos de HA são conjuntos de nós que colaboram para fornecer uma solução de alta disponibilidade para as VMs ou *containers*. Um nó no grupo pode assumir as VMs de outro nó em caso de falha. No nosso caso, foi criado um grupo englobando todos os nós com prioridades diferentes, como mostra a figura 5.41.

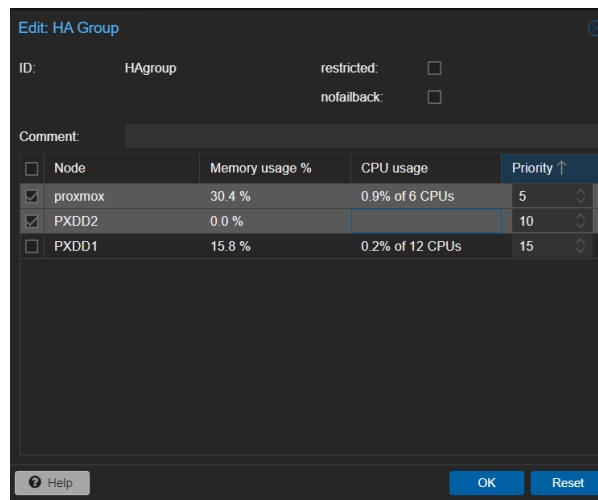


Figura 5.41: Criação de um grupo HA

De seguida, na aba de *resources*, adicionámos o *container* e a VM ao grupo HA criado, tal como está demonstrado nas 3 figuras seguintes: 5.42, 5.43 e 5.44.

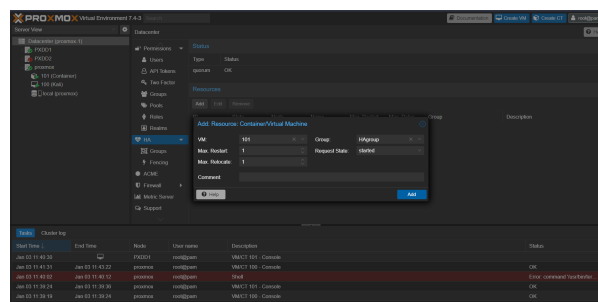


Figura 5.42: Adição de um *container* ao grupo

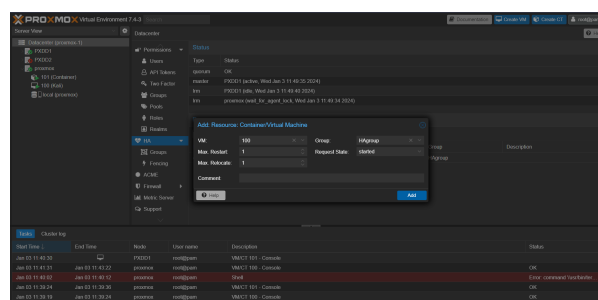


Figura 5.43: Adição de uma VM ao grupo

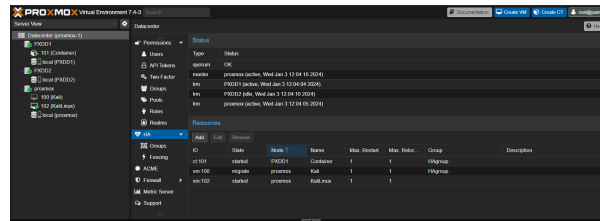


Figura 5.44: Resources

Depois de adicionado o *container* ao grupo, reparámos que houve uma migração para o nó com maior prioridade. Após algumas pesquisas, percebemos que é um comportamento normal. Quando um grupo é criado, é atribuído uma prioridade a cada nó no grupo. Essa prioridade determina qual nó é preferido para executar as VMs ou *containers*. Se atribuirmos uma prioridade mais alta a um determinado nó que não seja o nó onde está inicialmente alojado o *container*, o *proxmox* pode decidir migrar automaticamente o *container* para esse nó com prioridade mais alta, mesmo que o nó original esteja a funcionar corretamente. Em suma, foi possível concluir que a migração de *containers* entre nós num grupo de alta disponibilidade é influenciada pela prioridade configurada para cada nó.

5.12 Node failure

De forma a testar a resiliência do sistema criado foi feita a simulação de uma falha, introduzida manualmente, num dos nós do *cluster*. Para esta experiência foi decidido recorrer a outro processo, sem utilização de containers, para validar se a recuperação em situações de falha tinha melhor desempenho e de forma a permitir comparar as diferentes etapas.

5.12.1 Adição de *High-Availability*

Uma dos passos que foi logicamente diferente, foi aquando da criação de recursos HA, onde bastou selecionar a VM que se pretendia migrar em situação de falha do nó em que esta se encontrava, demonstrado na Figura 5.45.

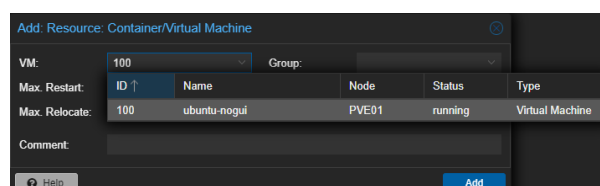


Figura 5.45: Resources HA

5.12.2 Validação *storage*

A ter em conta para que a experiência corra como planeado, no momento da criação da VM que vai ser alvo da migração, foi a seleção do *storage*. Visto que o *Ceph* utiliza os OSD para armazenamento compartilhado em casos de falha, para que possa obter a informação dos outros nós, foi preciso seleccionar a *pool* criada, como mostra a Figura 5.46

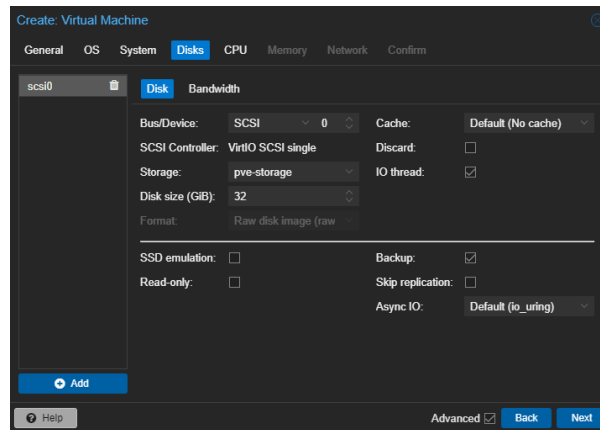


Figura 5.46: Escolha do storage na VM a migrar

De modo a forçar o encerramento do nó onde a VM se encontra, para simulação da falha, foi executado o comando da Figura 5.47. Após alguns minutos de espera o servidor *PVE01* é desligado, levando a que o sistema de monitorização da "saúde" desse nó detete que este se encontra em baixo.

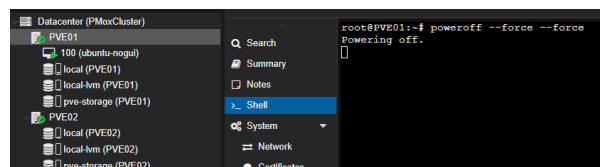


Figura 5.47: Configuração pré-falha no *PVE01*

Nesse momento, o nó de origem, onde anteriormente se encontrava a VM, prepara-se para a migração criando uma réplica dos dados da mesma num nó que se encontre saudável no cluster.

É então iniciada a migração da VM onde a replica anteriormente criada é gradualmente atualizada para um estado de *read-write*, permitindo que a VM faça a transição sem problemas para a utilização do armazenamento do nó de destino, como pode ser visto na Figura 5.48.

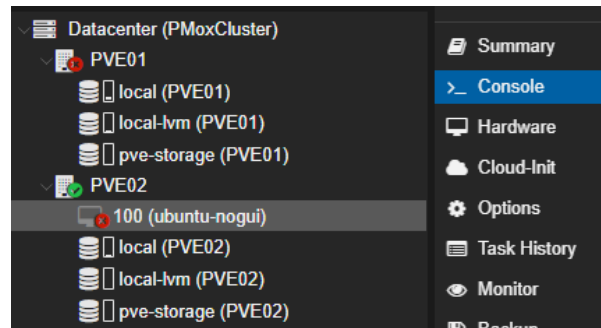


Figura 5.48: Configuração pós-falha no *PVE01*

Quando o processo é concluído, o nó de origem desmonta os dados da VM do seu armazenamento local, removendo efetivamente a VM desse nó.

Capítulo 6

Conclusões

Este capítulo apresenta uma reflexão de todo o trabalho desenvolvido para o cumprimento e conclusão do projeto B. São apresentados os objetivos alcançados e os conhecimentos acerca da plataforma sobre a qual trabalhamos, bem como as suas potencialidades.

6.1 Objetivos alcançados

Ao longo do projeto foi possível adquirir algum *know-how* sobre o *Proxmox* e todo o ambiente de virtualização que envolve, desde a criação de um *cluster* partilhado entre nós, máquinas virtuais, *containers*, HA, *ceph*, etc.

6.2 Problemas encontrados

A fase de experimentação é sempre um aspeto crucial para a compreensão por completo dos mecanismos de funcionamento de qualquer tecnologia e, portanto, é esperado que alguns erros surjam se não houver total entendimento de como trabalhar com esta.

Tais dificuldades passaram por manter a VM ativa quando esta é migrada de um nó para o outro, quando ocorre uma falha simulada. Uma das explicações para o sucedido deve-se ao facto da máquina que aloja os servidores não ter capacidade de processamento suficiente para que possam estar a correr os três nós a correr em simultâneo. Isto traduziu-se em algumas quedas nos servidores afetando estes processos.

6.3 Problemas em aberto

Um dos aspetos que pode ser considerado para trabalho futuro é a criação de um ficheiro no *file share* do *Ceph* para consolidar e aprofundar melhor o conhecimento relativo à migração.

6.4 Opinião acerca da tecnologia estudada

Foi extremamente útil a realização do projeto, uma vez que todo o seu ambiente, bem como os recursos que lhe estão relacionados são interessantes em contexto profissional. O *Proxmox* oferece uma solução integrada para virtualização de servidores, baseada em *containers Linux* e VMs, permitindo flexibilidade na implementação e gestão de ambientes virtualizados. Além disso, o *Proxmox* fornece uma interface *web* intuitiva que facilita a sua administração.

Referências

- [1] Cisco. *What Is High Availability?* URL: <https://www.cisco.com/c/en/us/solutions/hybrid-work/what-is-high-availability.html> (acedido em 28/09/2023).
- [2] Princeton Research Computing. *What is a cluster?* URL: <https://researchcomputing.princeton.edu/faq/what-is-a-cluster> (acedido em 28/09/2023).
- [3] Distro Domain. *Proxmox 8 cluster setup with ceph and HA*. URL: <https://www.youtube.com/watch?v=Mz-nXlqovLI> (acedido em 31/12/2023).
- [4] IBM. *Quorum disk*. URL: <https://www.ibm.com/docs/en/sanvolumecontroller/8.1.x?topic=details-quorum-disk> (acedido em 28/09/2023).
- [5] MRP. *Setting up Proxmox CLUSTER and STORAGE (Local, ZFS, NFS, CEPH) / Proxmox Home Server Series*. URL: <https://www.youtube.com/watch?v=a70Mi3bw0pQ> (acedido em 31/12/2023).
- [6] netapp. *High Availability Cluster: Concepts and Architecture*. URL: <https://bluexp.netapp.com/blog/cvo-blg-high-availability-cluster-concepts-and-architecture> (acedido em 31/12/2023).
- [7] Proxmox. *Cluster Manager*. URL: https://pve.proxmox.com/wiki/Cluster_Manager (acedido em 31/12/2023).
- [8] Proxmox. *Main Page*. URL: https://pve.proxmox.com/wiki/Main_Page (acedido em 28/09/2023).
- [9] pve.proxmox. *DeployHyper-ConvergedCephCluster*. URL: https://pve.proxmox.com/wiki/Deploy_Hyper-Converged_Ceph_Cluster (acedido em 31/12/2023).
- [10] VirtualizationHowto. *Proxmox 8 Cluster with Ceph Storage configuration*. URL: https://www.youtube.com/watch?v=-qk_P9SKYK4 (acedido em 31/12/2023).