MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RELATÓRIO DE ATIVIDADES DESENVOLVIDAS NO ESTÁGIO

RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTUDO DE TECNOLOGIAS DE DEVSECOPS EM UMA EMPRESA DE TECNOLOGIA DA INFORMAÇÃO

JOSUÉ CESAR RIBEIRO MENDONÇA

IPORÁ, GO
2024

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTUDO DE TECNOLOGIAS DE DEVSECOPS EM UMA EMPRESA DE TECNOLOGIA DA INFORMAÇÃO

JOSUÉ CESAR RIBEIRO MENDONÇA

LUCIANA RECART CARDOSO
Orientadora

Relatório de atividades de estágio desenvolvidas apresentado ao Instituto Federal Goiano – *Campus* Iporá, como requisito parcial para conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

IPORÁ, GO
Dezembro, 2024

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

Identificação da Produção Técnico-Científica (assinale com X)

[ ] Tese

[ ] Dissertação

[ ] Monografia – Especialização

[ ] Artigo - Especialização

[X] TCC - Graduação (Estágio)

[ ] Artigo Científico

[ ] Capítulo de Livro

[ ] Livro

[ ] Trabalho Apresentado em Evento

[ ] Produção técnica. Qual: Relato de experiência

Nome Completo do Autor: Josué Cesar Ribeiro Mendonça
Curso: Tecnologia em Análise e Desenvolvimento de Sistemas
Título do Trabalho: Relatório de atividades desenvolvidas: estudo de tecnologias de DevsEcops em uma empresa de tecnologia da informação

Restrições de Acesso ao Documento [Preenchimento obrigatório]

Documento confidencial: [X] Não [ ] Sim, justifique: _____

Informe a data que poderá ser disponibilizado no RIIF Goiano: 04/03/2024

O documento está sujeito a registro de patente? [ ] Sim [X] Não

O documento pode vir a ser publicado como livro? [ ] Sim [X] Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

1. O documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;

2. Obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;

3. Cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Iporá, 5 de dezembro de 2024

Josué Cesar Ribeiro Mendonça

Assinado eletronicamente pelo o Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

Luciana Recart Cardoso

Assinatura eletrônica do(a) orientador(a)

Documento assinado eletronicamente por:

- **Luciana Recart Cardoso**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 05/12/2024 19:42:00.
- **Josué César Ribeiro Mendonça**, **2018105210430070 - Discente**, em 05/12/2024 19:45:58.

Este documento foi emitido pelo SUAP em 05/12/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifgoiano.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 658940
Código de Autenticação: 524e45a828

INSTITUTO FEDERAL GOIANO
Campus Iporá
Av. Oeste, Parque União, 350, Parque União, IPORA / GO, CEP 76.200-000
(64) 3674-0400

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 128/2024 - GE-IP/CMPIPR/IFGOIANO

ATA DA SESSÃO DE JULGAMENTO DO TRABALHO DE CURSO

DE JOSUÉ CESAR RIBEIRO MENDONÇA

Aos quatro dias, do mês de dezembro de dois mil e vinte e quatro, às vinte e uma horas e quatro minutos, em sessão pública, a banca examinadora designada na forma regimental pela Coordenação do Curso para julgar o trabalho de curso intitulado **"RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTUDO DE TECNOLOGIAS DE DEVSECOPS EM UMA EMPRESA DE TECNOLOGIA DA INFORMAÇÃO "**, apresentada pelo acadêmico Josué Cesar Ribeiro Mendonça, como parte dos requisitos necessários à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas. A banca examinadora foi presidida pela orientadora do trabalho de curso, professora **Ma. Luciana Recart Cardoso**, tendo como membros avaliadores: o professor **Dr. Thamer Horbylon Nascimento** e o professor **Ma. Lais Candido Rodrigues da Silva Lopes**. Aberta a sessão, o acadêmico expôs seu trabalho. Em seguida, foi arguido pelos membros da banca e:

(X) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** do acadêmico, sem restrições.

( ) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** do acadêmico, **condicionada a satisfazer as exigências** listadas na Folha de Modificação de Trabalho de Curso anexa à presente ata, no prazo máximo de 80 (oitenta) dias, a contar da presente data, ficando o professor orientador responsável por atestar o cumprimento dessas exigências.

( ) não tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **reprovação** do acadêmico.

Conforme avaliação individual de cada membro da banca, será atribuída a nota nove (9,5) para fins de registro em histórico acadêmico.

Os trabalhos foram encerrados às 21 horas e 40 minutos. Nos termos do Regulamento do Trabalho de Curso do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal Goiano – Campus Iporá, lavrou-se a presente ata que, lida e julgada conforme, segue assinada pelos membros da banca examinadora.

Assinado eletronicamente

Luciana Recart Cardoso, Ma. Orientadora

Thamer Horbylon Nascimento, Dr.

Lais Candido Rodrigues da Silva Lopes, Ma.

Documento assinado eletronicamente por:

- **Luciana Recart Cardoso**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 04/12/2024 21:44:04.
- **Thamer Horbylon Nascimento**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 04/12/2024 21:46:09.
- **Lais Candido Rodrigues da Silva Lopes**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 04/12/2024 21:46:47.

Este documento foi emitido pelo SUAP em 04/12/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifgoiano.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 658343
Código de Autenticação: 72bafc985e

INSTITUTO FEDERAL GOIANO
Campus Iporá
Av. Oeste, Parque União, 350, Parque União, IPORA / GO, CEP 76.200-000
(64) 3674-0400

SUMÁRIO

1 – INTRODUÇÃO

Neste relatório serão abordadas atividades, conteúdos, e aprendizados os quais foram desenvolvidos durante o estágio de *DevSecOps* na empresa Compass.UOL. Cada um dos temas será aprofundado, sendo que as atividades eram as formas com as quais atuávamos como bolsistas. Serão enfatizados aprendizados que contribuíram para o meu crescimento, os benefícios trazidos pelo estágio à minha carreira.

O estágio foi um programa de bolsas criado para formar profissionais em uma área de demanda da empresa contratante. Precisavam de mais profissionais de *DevSecOps*, o objetivo era oferecer estas vagas a instituições de ensino e com isso convocar estudantes como bolsistas, os quais passariam por uma série de treinamentos. Foi determinante tanto o preparo comportamental relacionado a forma de trabalho, seguindo *sprints* juntamente das metodologias ágeis aplicadas, como ter um perfil proativo, e dominar as práticas e técnicas entregues através de cursos para nos profissionalizarmos na área. Contudo, no final desta preparação os bolsistas que se desenvolvessem conforme o interesse da companhia, teriam a chance de ter uma oportunidade de contratação.

2 – CARACTERIZAÇÕES DA EMPRESA E DADOS DO ESTÁGIO

2.1 – Caracterizações da empresa

A empresa hoje chamada Compass UOL, chamava-se Compasso e foi fundada no ano de 1995, em Passo Fundo - RS, em 2013.

A Compasso foi adquirida pela UOL Diveo, as duas companhias se dividiram em duas missões diferentes, enquanto a UOL Diveo, agora UD Tecnologia, atua com foco em controle de *data centers*, a Compass UOL opera no setor de Transformação Digital, ajudando seus clientes com consultoria na estruturação de soluções para diversos casos diferentes focando em *Cloud* e *IA* (COMPASS, 2020).

A Compass UOL é uma empresa que foca na entrega de serviços de alta qualidade gerando impactos positivos na sociedade, provendo valores a seus clientes e parceiros entregando serviços e produtos eficazes, com qualidade e segurança. Trabalha com foco no cliente, sempre visando inovação e agilidade, qualidade e melhoria contínua, segurança e conscientização e impacto positivo (COMPASS, 2021).

2.2 - Dados do estágio

O estágio foi realizado desde o dia 6 de agosto de 2021 e 12 de novembro de 2021, totalizando uma carga horária de 240 horas. O acompanhamento do estágio foi realizado pelas supervisoras internas Adriana Rodrigues e Andreza Santarém, que também regiam as reuniões diárias. Já a supervisão externa, foi realizada pela professora e orientadora Luciana Recart Cardoso.

Durante o período de estágio, foram organizadas jornadas de 4h diárias, onde teríamos responsabilidades a prestar. A princípio, teríamos diariamente ao início da carga horária, uma *Daily Standup*, que no meio é simplesmente chamada de *daily*, onde foi o primeiro contato de muitos de nós com a metodologia ágil *Scrum* no mundo profissional. Após a *daily*, partíamos para a nossa jornada de trabalho, que era dividida em *sprints*, que são períodos fixos nos quais tínhamos um plano de entrega definido para o fim do prazo. Geralmente as *sprints* são

realizadas em intervalos de duas semanas, durante o estágio esse foi o tamanho definido para as *sprints*, onde o nosso objetivo era consumir certos módulos de cursos dentro do período e entregar projetos para mostrar a proficiência adquirida e sermos avaliados. A avaliação era interna, havia sempre verificações e acompanhamento de nosso desempenho.

Durante a pandemia, cerca de 46% das empresas adotaram o modelo remoto de trabalho, a Compass UOL desde 2020, com a declaração de pandemia de COVID-19 pela Organização Mundial da Saúde (OMS) (MELLO, 2020).

A partir de março de 2021, a empresa tem operado 100% no modelo remoto, seguindo até os dias de hoje, em 2024 (COMPASS, 2021). O estágio foi realizado durante o período do segundo semestre de 2021, no modelo remoto, onde para contato utilizávamos da ferramenta *Microsoft Teams*, sendo tanto para contato entre colegas e nossos gestores, tanto para nossas reuniões diárias, as *Daily Stand Ups*. Cada estagiário recebeu um *e-mail* institucional para uso durante o programa.

Para os cursos, foram fornecidas contas com acesso à plataforma de ensino Alura, e registrávamos as horas de trabalho em uma plataforma interna da companhia de apropriação de horas e atividades. Fomos separados em três grupos diferentes para não estender o tempo das *daily stand ups,*lém do necessário, e sempre que havia um resultado a ser entregue, éramos separados em equipes de três integrantes.

2.3 - Principais metodologias e ferramentas

No decorrer do estágio tivemos contato com algumas metodologias, ferramentas e tecnologias do mercado, mas algumas se destacam por sua importância e impacto no meio de abrangência do estágio.

2.3.1 - Introdução à metodologia ágil SCRUM

As *Dailys* e *Sprints* são parte do *framework SCRUM*, que é uma metodologia ágil focada em acelerar os projetos e aumentar a colaboração das equipes através de regras definidas pelo *framework*. Onde se incentiva o aprendizado a partir das experiências e faz com que os

profissionais avaliem sucessos e falhas, assim promovendo uma melhoria contínua (SCHWABER; SUTHERLAND, 2020).

As *Sprints* são períodos pré-definidos de até um mês, no qual a equipe trabalha para entregar uma quantidade específica de trabalho, ou um resultado esperado (ATLASSIAN, 2024).

Durante o programa, as sprints foram definidas para períodos de 2 semanas. No início de cada sprint, era necessário planejar a entrega esperada ao seu fim e definir como seria realizado.

Já as *Daily Stand Ups*, são reuniões diárias, idealmente de no máximo 15 minutos, onde o foco é ser objetivo e claro, ocupando o mínimo de tempo possível, sempre focando em dizer, o que foi realizado desde a última reunião, o que há em pretensão para ser realizado no dia e se existe algum obstáculo para que possa atingir as demandas ou a *sprint.* A partir disto, discorríamos sobre estas perguntas um por vez, caso alguém tivesse algum impedimento, como equipe deveríamos fazer o possível para apoiar o indivíduo com problemas (SCHWABER; SUTHERLAND, 2020).

No fim de cada *sprint*, era feita uma reunião retrospectiva, na qual eram avaliadas as atividades realizadas, dificuldades, sucessos e falhas, onde os resultados serviam de aprendizado para as próximas etapas.

2.3.2 - Implementação da cultura DevSecOps

O projeto de estágio foi baseado em profissionalizar-nos como *DevSecOps*, vindo do *DevOps*, uma cultura que visa aproximar e melhorar a comunicação entre as áreas de desenvolvimento. Desse modo, tornando-se a intersecção entre o desenvolvedor e a área de operações, onde os produtos são entregues a ambientes produtivos (ATLASSIAN; AWS, 2024).

O profissional de *DevOps* controla os ciclos de desenvolvimento de um produto, criando automações e gerenciando esteiras de integração e entrega contínua, conhecidas como CI/CD (*Continuous Integration / Continuous Delivery*), que visam automatizar os processos de integração de código, por parte dos desenvolvedores. Na sequência, passando por testes de integração, análises de vulnerabilidade e são feitos *builds* das aplicações, assim inteirando o

processo de CI. Se tudo der certo e todos os testes forem bem-sucedidos, após estas etapas a aplicação com seu build realizado, inicia-se a etapa de CD. No processo de CD, a aplicação é empacotada em uma imagem versionada e enviada para um repositório de imagens e aplicada em ambiente de desenvolvimento, homologação ou produtivo dependendo do estado do código integrado (POULTON, 2021).

Isso tudo é uma esteira automatizada focada em facilitar a integração de código e a velocidade de entrega, mas os profissionais de *DevOps* vão além de criar esteiras, atuam com o desenvolvimento de automações para migrações e alterações em massa. Trabalham com provisionamento de recursos *on premise (data centers* proprietários) ou em *Cloud*, também atuam com a criação de *infrastructure as code* (IAC), dentre outras práticas situacionais (ATLASSIAN, 2024).

Já o *DevSecOps* atua com todo o resto, com um foco extra em segurança, com práticas como o *shift-left*, uma prática focada em aumentar a segurança e o número e qualidade de testes nas primeiras etapas de desenvolvimento. Visando a esteira de CI/CD por exemplo, é onde o termo *shift-left* ganha evidência, quanto mais à esquerda estiverem os testes nas sequências de etapas da esteira, maior o nível dessa prática aplicada no projeto. Isso ajuda para que um software desenvolvido dentro dessas metodologias, seja melhor estruturado, mais seguro e com poucos problemas ou falhas (ATLASSIAN; AWS, 2024).

2.3.3 - Linux

Linux é um sistema operacional de código aberto, criado e mantido com contribuições de sua comunidade. O Linux é utilizado em servidores, *desktops* e dispositivos incorporados e nenhuma empresa possui sua propriedade, é desenvolvido coletivamente sob licenças como a *General Public License* (GPL) (BRESNAHAN; BLUM, 2020).

Existem várias distribuições do Linux, cada uma adaptada para diferentes necessidades, como Red Hat, Ubuntu e Oracle Linux. Essas distribuições podem oferecer suporte técnico, atualizações e recursos específicos, podendo variar seu desempenho e cada tipo de Linux tem seu próprio sistema de gerenciamento de pacotes dependendo de sua versão ou imagem base, como Debian com ‘apt’, Fedora com ‘yum’.

O Linux é robusto e seguro, além de capaz de atualizar e modificar partes importantes do sistema sem sequer precisar reiniciar o computador, o que é muito importante em servidores em ambientes corporativos. É um sistema totalmente gratuito e acessível, leve que pode rodar eficientemente em máquinas mais antigas (BRESNAHAN; BLUM, 2020).

2.3.4 - Containers

Containers são uma tecnologia que possibilita o agrupamento de aplicações em ambientes isolados. Essas aplicações são executadas de maneira diferente das máquinas virtuais, que, para emular um sistema operacional (SO), precisam também emular seu *hardware*, consumindo assim muito mais recursos do sistema. Já os *containers* compartilham o *kernel* do sistema, o que elimina a necessidade de emular o *hardware*, permitindo o compartilhamento de recursos do *host*. Dessa forma, apenas o SO é emulado, utilizando os recursos disponibilizados pelo *host*.

O sistema operacional dos containers contém o mínimo necessário para executar as aplicações, tornando-se bastante performático, com um consumo próximo ao que se teria ao rodar as aplicações diretamente na máquina (DESCOMPLICANDO DOCKER, 2024).

O uso de containers resulta em um consumo maior do que a execução direta na máquina *host*, porém, sua vantagem é virtualização que fornece ambientes isolados e sem conflitos com outras aplicações ou recursos. Funciona em qualquer ambiente e pode executar sistemas operacionais tanto pré-definidos quanto personalizados. Essa singularidade permite que uma mesma aplicação tenha comportamentos diferentes em máquinas distintas, pois cada uma delas possui seu próprio espaço de variáveis de ambiente. Além disso, a virtualização elimina problemas de instalação de dependências, mantendo assim um ambiente livre de inconsistências, o que elimina o famoso dito “Mas na minha máquina funciona!” (DESCOMPLICANDO DOCKER, 2024).

2.3.5 - Docker

Docker é uma tecnologia que permite criar e gerenciar *containers*, proporcionando uma maneira modular e eficiente de desenvolver, implantar e migrar aplicações. O projeto é mantido

por uma comunidade *open source*, com o apoio da empresa Docker Inc., que oferece suporte e segurança como soluções pagas (DESCOMPLICANDO DOCKER, 2024).

Docker utiliza o *kernel* do sistema da máquina *host* para isolar processos, permitindo que diferentes aplicativos sejam executados em um mesmo sistema sem conflitos. Também oferece vantagens como modularidade, controle de versões e rápida implantação, permitindo que novas aplicações sejam criadas e disponibilizadas em segundos (DESCOMPLICANDO DOCKER, 2024).

Quando a quantidade de *containers* aumenta pode se tornar complexo gerenciar tudo via Docker, necessitando de ferramentas de orquestração como o Kubernetes. Apesar de ser eficiente, o Docker tem algumas limitações de segurança, pois compartilha o *kernel* da máquina *host* com os *containers*, o que pode gerar vulnerabilidades (DESCOMPLICANDO DOCKER, 2024).

2.3.6 - Kubernetes

Kubernetes, comumente chamado de “k8s”, um apelido dado pela comunidade, onde “K” representa a primeira letra, “8” o número de letras até a última letra "S”, é uma plataforma *open source* de orquestração de containers. Seu foco é automatizar a implantação, o gerenciamento e o dimensionamento de aplicações conteinerizadas (ATLASSIAN; AWS, 2024).

Desenvolvido inicialmente pelo Google e posteriormente disponibilizado como *open source*, o Kubernetes capacita desde usuários comuns até grandes organizações a implementar e escalar aplicações de forma eficiente, tanto em ambientes *on-premises* quanto em nuvem. Coordena diversas tarefas, como descoberta de serviços, balanceamento de carga e autocorreção, resultando em alta disponibilidade. O Kubernetes se tornou a solução mais popular no seu escopo, superando outras opções devido à sua ampla funcionalidade e ao suporte robusto que diversos provedores de nuvem oferecem (POULTON, 2021).

2.3.7 - AWS

Amazon Web Services (AWS), é uma plataforma com mais de 200 serviços em nuvem, distribuídos em seus diversos *data centers* pelo mundo. Possui milhões de clientes, e tem seu

foco em reduzir os custos de computação de seus clientes. Seus *datacenters* posicionados de forma geograficamente estratégica oferecem estabilidade em casos de indisponibilidade, trazendo resiliência para seus clientes (AWS, 2024).

Algumas das principais áreas de atuação da empresa hoje são infraestrutura, computação em nuvem, armazenamento de dados, *machine learning* e análise da dados (AWS, 2024).

3 – ATIVIDADES DESENVOLVIDAS

Durante o período de estágio, foram divididas as atividades a serem cumpridas em listas de cursos a serem consumidos na plataforma de cursos Alura, e no fim de alguns dos blocos de cursos dentro do tema de DevSecOps, eram aplicados desafios para serem resolvidos em grupo.

Quadro 1 - Lista de cursos 01: Linux.

CURSO	**TEMA**	**HORAS**
Entendendo o Linux	Linux I - Conhecendo e utilizando o terminal	4h
Entendendo o Linux	Linux II - Programas, processos e pacotes	8h
Certificação LPI Linux *Essentials*	*Evolution and Distributions*	8h
Certificação LPI Linux *Essentials*	*Open-Source Software and Licensing*	8h
Certificação LPI Linux *Essentials*	*Command Line Basics*	8h
Certificação LPI Linux *Essentials*	*Using the command line to get help*	8h
Certificação LPI Linux *Essentials*	*Directories and Listing Files and managing files*	8h
Certificação LPI Linux *Essentials*	*Archiving Files on the Command Line*	5h
Certificação LPI Linux *Essentials*	*Searching and Extracting Data from Files*	4h
Certificação LPI Linux *Essentials*	*Turning Commands into a Script*	5h
Certificação LPI Linux	*Understanding Computer Hardware*	4h

Essentials		
Certificação LPI Linux *Essentials*	*Where Data is Stored*	4h
Certificação LPI Linux *Essentials*	*Your Computer on the Network*	4h
Certificação LPI Linux *Essentials*	*Security and File Permissions*	4h

- **Entendendo o Linux: Linux I - Conhecendo e utilizando o terminal:** Neste módulo, foi feita uma introdução ao terminal Linux, uma ferramenta essencial para usar o sistema. Foram ensinados comandos básicos para navegar pelos arquivos, como 'ls' para listar diretórios e 'cd' para mudar de pasta. Além disso, foram abordadas as operações de criação, remoção e edição de arquivos, formando uma base sólida para o uso do Linux.

- **Entendendo o Linux: Linux II - Programas, processos e pacotes:** Este módulo focou no gerenciamento de programas e processos no Linux. Foram apresentados métodos para instalar software usando gerenciadores de pacotes como 'apt' e 'yum', além de monitorar processos em execução e finalizar aqueles que estavam consumindo muitos recursos. Essas habilidades são fundamentais para manter o sistema funcionando corretamente.

- **Certificação LPI Linux *Essentials: Evolution and Distributions:*** Aqui, foram discutidos a evolução do Linux e suas principais distribuições. A comparação entre distribuições populares, como Ubuntu, Fedora e Debian, e ajuda a entender como escolher a mais adequada para diferentes situações. Esse conhecimento é fundamental para saber fazer a escolha do ambiente ideal para diferentes projetos.

- **Certificação LPI Linux *Essentials: Open-Source Software and Licensing:*** Neste módulo, o conceito de software livre e as diferentes licenças, como GPL e MIT *(Massachusetts Institute of Technology)*, foram abordadas. A compreensão dos direitos e responsabilidades ao usar e compartilhar software *open source*, isto mostra a importância do trabalho em equipe e como a colaboração em projetos de comunidades podem dar acesso a ferramentas sólidas e gratuitas.

- **Certificação LPI Linux *Essentials: Command Line Basics:*** Foram exploradas mais a fundo ferramentas de linha de comando. A sintaxe dos comandos e o uso de redirecionamento e *pipes* foram ensinados, permitindo juntar e manipular dados de forma eficiente. Essas habilidades são essenciais para automatizar tarefas e auxiliar no aumento de produtividade.

- **Certificação LPI Linux *Essentials: Using the command line to get help:*** Neste módulo, foram apresentadas ferramentas de ajuda, como 'man' e '--help', que são úteis para encontrar informações sobre comandos e suas opções. Saber como consultar a documentação é muito valioso na hora de resolver problemas ou aprender a utilizar uma nova ferramenta.

- **Certificação LPI Linux *Essentials: Directories and Listing Files and Managing Files:*** A estrutura do sistema de arquivos Linux foi detalhada, com ênfase em uma gestão mais eficiente dos arquivos. Discorreu-se sobre o uso de comandos para listar e manipular diretórios, o que ajuda a melhorar a organização dos dados.

- **Certificação LPI Linux *Essentials: Archiving Files on the Command Line:*** Foi ensinado o uso de ferramentas de compressão, como 'tar' e 'gzip', para compactar e descompactar arquivos. Essa habilidade é útil ao lidar com grandes volumes de dados ou fazer backups.

- **Certificação LPI Linux *Essentials: Searching and Extracting Data from Files:*** Este módulo mostrou como buscar dados dentro de arquivos usando comandos como 'grep' e 'find'. Saber localizar informações rapidamente é uma habilidade valiosa em muitas situações práticas.

- **Certificação LPI Linux *Essentials: Turning Commands into a Script:*** Neste módulo, a criação de scripts em Bash foi abordada. Foi demonstrada a estruturação de scripts com variáveis, loops e condicionais, mostrando como automatizar tarefas repetitivas e ganhar eficiência em atividades diárias.

- **Certificação LPI Linux *Essentials: Understanding Computer Hardware:*** Aqui, foram discutidos os componentes de hardware e como o Linux interage com estes. Entender os

requisitos de hardware para diferentes aplicações ajuda a ter uma visão melhor do funcionamento do sistema.

- **Certificação LPI Linux *Essentials: Where Data is Stored:*** A estrutura de armazenamento no Linux foi analisada, incluindo sistemas de arquivos e gestão de espaço em disco. Esse conhecimento é fundamental para garantir a eficiência na navegação no sistema.
- **Certificação LPI Linux *Essentials: Your Computer on the Network:*** Neste módulo, foram abordados conceitos básicos de redes, como IP, DHCP e DNS. Mostra como ferramentas de rede no Linux são essenciais para entender a comunicação entre computadores em rede.

- **Certificação LPI Linux *Essentials: Security and File Permissions:*** Por fim, discutiu-se a segurança no Linux e a gestão de permissões de arquivos e diretórios. Foi apresentada a importância de se proteger dados e controlar quem tem acesso a determinados arquivos ou diretórios.

3.1 – Exercício de fixação Linux

Após o término da primeira lista de cursos, onde obtivemos conhecimentos fundamentais sobre Linux, colocamos nossos aprendizados à prova em um exercício que foi executado em grupos de 3 integrantes cada. Era necessário criar duas máquinas Linux, uma com PHP e Apache e outra com NGINX, onde a máquina com NGINX deveria haver um site HTML que através dele e de algumas configurações de rede entre as máquinas virtuais, fosse possível requisitar o index.html e ver as informações do PHP da máquina Apache.

O SO escolhido para o exercício foi o Linux Ubuntu Server, versão LTS 20.04.3, as máquinas foram criadas utilizando VirtualBox, foram criadas as duas máquinas, sendo máquina-01 com Apache e PHP, e máquina-02 com NGINX.

Após criação das máquinas no VirtualBox, iniciamos suas configurações.

3.1.1 – Configuração máquina-01

- **Instalação e configuração do Apache**

1. Para instalar o Apache, por padrão, a lista de repositórios do apt não continha o Apache. Por isso, é necessário atualizar:

sudo apt-get update: atualiza a lista de pacotes disponíveis.

2. Após atualizar a lista de repositórios do apt, pudemos baixar o Apache:

sudo apt-get install apache2: instala o servidor Apache.

3. Depois de instalar o Apache, é necessário habilitar o *firewall* para liberar o acesso ao Apache e configurá-lo para iniciar automaticamente na inicialização da máquina.

sudo ufw enable: ativa o *firewall*.

sudo ufw allow 'Apache': permite conexões ao Apache.

4. Posteriormente pudemos validar se a alteração foi bem-sucedida através do seguinte comando:

sudo ufw status: mostra o status das aplicações no *firewall* e suas permissões.

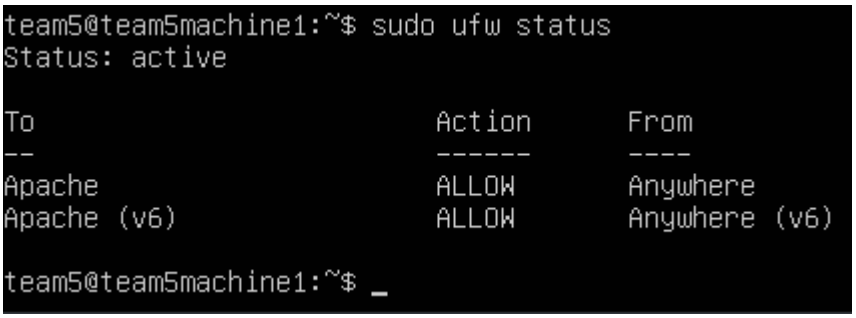


Figura 1 - Retorno de status do firewall.

O comando deve retornar o status das aplicações no *firewall* e suas permissões, na Figura 1 pôde-se validar na coluna "*Action*" que o Apache possui permissão habilitada no *firewall*.

5. Finalmente, deverímos verificar se o Apache está sendo executado. Pudemos fazer isso com o seguinte comando:

sudo systemctl status apache2: verifica o status do Apache.

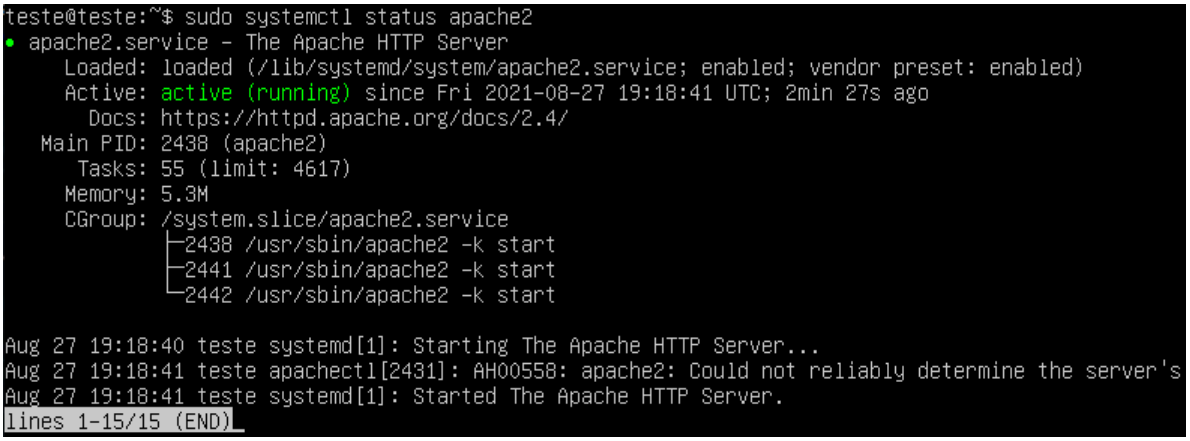


Figura 2: Retorno status do apache2.

O comando retorna o status de execução no sistema. Pudemos validar na figura 2 no status "*Active*" como "*active (running)*".

- **Instalação e configuração do PHP**

1. Após a instalação bem-sucedida do Apache, se pode seguir para o próximo passo: a instalação do PHP na máquina-01. Para isso, precisamos instalar algumas dependências para o seu funcionamento:

sudo apt install ca-certificates apt-transport-https software-properties-common: instala dependências.

2. Depois disso, adicionamos o repositório que contém a versão do PHP que queríamos instalar:

sudo add-apt-repository ppa:ondrej/php: adiciona o repositório do PHP.

3. Com isso, precisamos atualizar as listas de pacotes, e após a atualização, instalar a versão desejada:

sudo apt update && sudo apt install php8.0 libapache2-mod-php8.0: instala o PHP e o módulo do Apache.

4. Após a instalação do PHP, foi necessário reiniciar o serviço do Apache:

sudo systemctl restart apache2: reinicia o Apache.

5. Para validar a instalação do PHP, podemos executar o comando abaixo, que deve retornar a versão instalada:

php -v: verifica a versão do PHP.

6. Para utilizar o PHP, basta adicionar seus arquivos .php no diretório /var/www/html/.

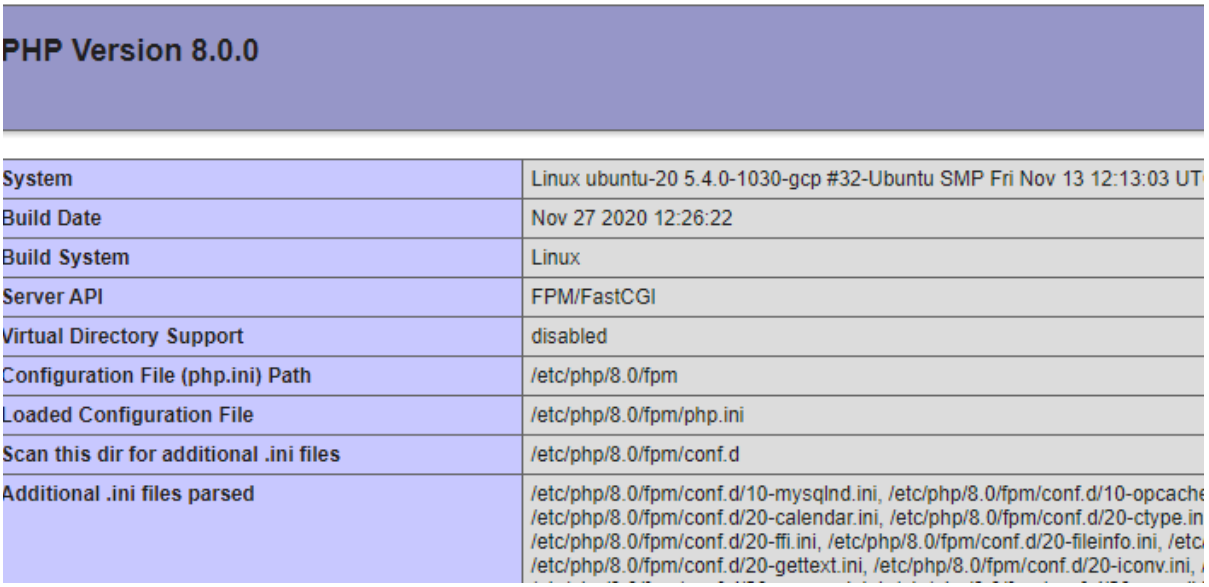
sudo vim /var/www/html/info.php: cria um arquivo PHP.

7. Adicione o seguinte conteúdo presente na Figura 3:\

```php
<?php
phpinfo();
?>
```

Figura 3: Código PHP.

8. Agora bastava acessar o endereço do seu PHP, que será o IP de sua máquina, exemplo “http://server-ip/info.php”, onde “server-ip” é o seu IP, essa chamada deve retornar uma página com as informações do seu PHP instalado, como mostrado na Figura 4:



PHP Version 8.0.0

System	Linux ubuntu-20 5.4.0-1030-gcp #32-Ubuntu SMP Fri Nov 13 12:13:03 UT
Build Date	Nov 27 2020 12:26:22
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.0/fpm
Loaded Configuration File	/etc/php/8.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.0/fpm/conf.d
Additional .ini files parsed	/etc/php/8.0/fpm/conf.d/10-mysqlnd.ini, /etc/php/8.0/fpm/conf.d/10-opcache /etc/php/8.0/fpm/conf.d/20-calendar.ini, /etc/php/8.0/fpm/conf.d/20-ctype.in /etc/php/8.0/fpm/conf.d/20-ffi.ini, /etc/php/8.0/fpm/conf.d/20-fileinfo.ini, /etc /etc/php/8.0/fpm/conf.d/20-gettext.ini, /etc/php/8.0/fpm/conf.d/20-iconv.ini,

Figura 4: Informações do PHP.

- **Configuração de disco**

1. Após configurar o PHP, um dos desafios do exercício era após a VM criada, adicionar um disco novo nela de 500mb, e montá-lo particionado na máquina-01, em formato ext4. Para isso adicionamos o disco pelo gerenciador de discos do VirtualBox, e iniciamos novamente a máquina, para ver os discos presentes, podemos rodar o comando:

sudo fdisk -l: lista os discos.

Este comando deve retornar algo semelhante a Figura 5:

```
Disk /dev/sdb: 500 MiB, 524288000 bytes, 1024000 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/mapper/ubuntu--vg-ubuntu--lv: 8.102 GiB, 9659482112 bytes, 18866176 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Figura 5: Listagem de discos.

2. Nosso novo disco é o /dev/sdb, precisamos particioná-lo para podermos utilizá-lo, para isso podemos selecioná-lo utilizando o seguinte comando:

sudo fdisk /dev/sdb: Inicia a configuração para o disco.

Com isto, irá iniciar o gerenciador de discos fdisk com o disco /dev/sdb selecionado como mostra na Figura 6:

Figura 6: Modo de configuração de disco.

3. Para criar uma nova partição no disco pressiona-se "N", em seguida aparecerão duas opções, "P" para "*primary"* ou "E" para "*extended*". Como não precisaremos fazer boot na partição, seleciona-se "extended" pressionando a tecla "E" e confirmando com a tecla "Enter", como mostra na Figura 7:

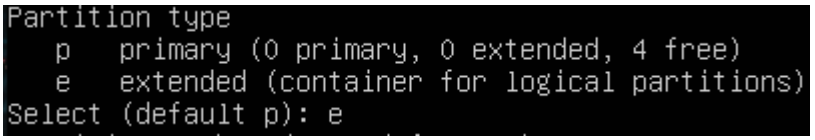


Figura 7: Escolhendo o tipo de partição.

4. Em seguida podemos escolher um número para a partição, ou mesmo não informar um número, desta forma por padrão, será selecionado o número 1, e pressionar "Enter". Depois será requisitado o setor onde inicia o disco, e ainda onde encerra, basta pressionar Enter em ambas as opções sem colocar nenhum valor para criar desde o início até o fim, assim como demonstra a Figura 8:

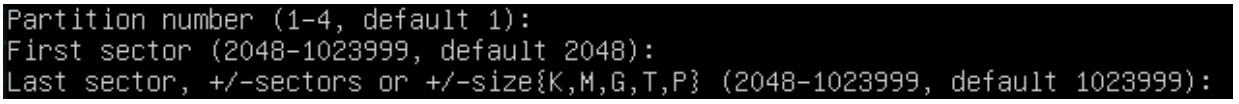


Figura 8: Dimensionamento de partição.

5. Ao inserir todos estes dados, ou como no nosso caso, deixando os valores vazios, o projeto de partição é criado, mas ainda não foi escrito no disco. Para realmente particionar o disco com as configurações feitas, é necessário escrever as alterações no disco. Para isto, utilizamos a tecla “W”. Na figura 9 podemos verificar que a partição foi criada com sucesso.

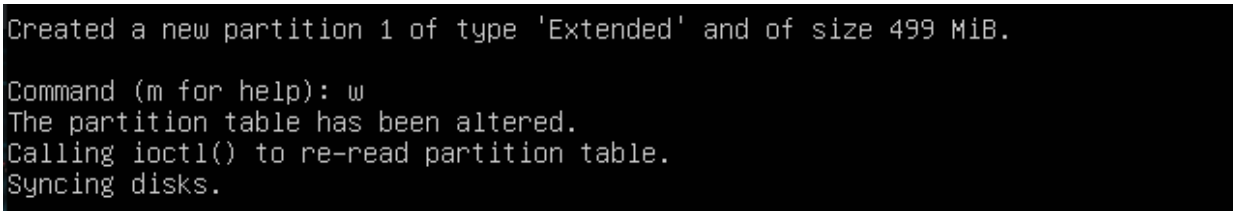


Figura 9: Particionamento concluído.

6. Depois de escrever as configurações no disco, a aplicação fdisk irá encerrar automaticamente. Com isso nosso disco foi particionado, agora precisamos formatá-lo em ext4, para isso precisamos executar o seguinte comando:

sudo mkfs.ext04 /dev/sdb: Formata o disco em ext4.

7. Agora com o disco formatado, para podermos utilizá-lo precisamos montá-lo em um diretório de nosso sistema, para isso vamos criar um diretório e depois apontar o disco para usá-lo como ponto de montagem.

sudo mkdir /u01: Cria o diretório para montagem do disco.

sudo mount /dev/sdb /u01: Monta o disco no diretório /u01.

8. Agora temos o disco montado no diretório “/u01”, mas sempre que a máquina for reiniciada ou desligada, será necessário montar manualmente a partição no diretório novamente, para resolver isto e automaticamente montar esta partição no diretório “/u01”. É necessário adicionar esta partição ao arquivo “/etc/fstab”, que é responsável pela montagem automática dos discos na inicialização do sistema. Basta adicionar isto ao arquivo: “/dev/sdb /u01 ext4 defaults 0 0”.

9. Nesse momento, se tudo foi feito corretamente, ao reiniciar a máquina, o disco deve estar montado no diretório "/u01". Para verificar se "/dev/sdb" foi montado em "/u01" basta se executar:

findmnt -S /dev/sdb: Lista os discos montados pelo sistema.

- **Criando diretórios do servidor**

1. Agora que temos os diretórios /u01, devemos criar os seguintes diretórios e arquivos:

/u01/apache/www/imagens

/u01/apache/www/imagens/< LOGO DA COMPASSO >

/u01/apache/www/index.html <LOGO COMPASSO E LINK P/ INFOS_PHP >

/u01/apache/www/infos_php.php < USANDO A FUNÇÃO phpinfo() >

- **Configuração das portas do servidor**

1. Se tentarmos acessar "localhost:8070", não será possível, pois por padrão o Apache usa a porta "80" e o diretório "/var/www/html". Para configurar nosso servidor para rodar o diretório "/u01/apache/www", vamos fazer algumas configurações no apache.

Os arquivos de configuração do apache ficam na pasta "/etc/apache2", para configurar a porta, vamos editar o arquivo "ports.conf". Dentro do arquivo, será encontrado algo parecido com a Figura 10:

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
```

Figura 10: Configurações padrão Apache.

2. Por padrão, podemos ver que a porta escutada é a “80”, como precisamos usar apenas a “8070”, basta alterar a porta de “80” para “8070”, igual a Figura 11:

Figura 11: Configurações Apache realizadas.

3. Após alterar a porta, é necessário salvar, após salvar, sair do arquivo. Agora, devemos alterar o arquivo de configuração de sites habilitados “000-default.conf”, localizado em “/etc/apache2/sites-enabled”. Devemos alterar “:80” na primeira linha para “:8070”, e em “DocumentRoot”, onde se encontra “/var/www/html”, devemos alterar para “/u01/apache/www”, como demonstrado na Figura 12:

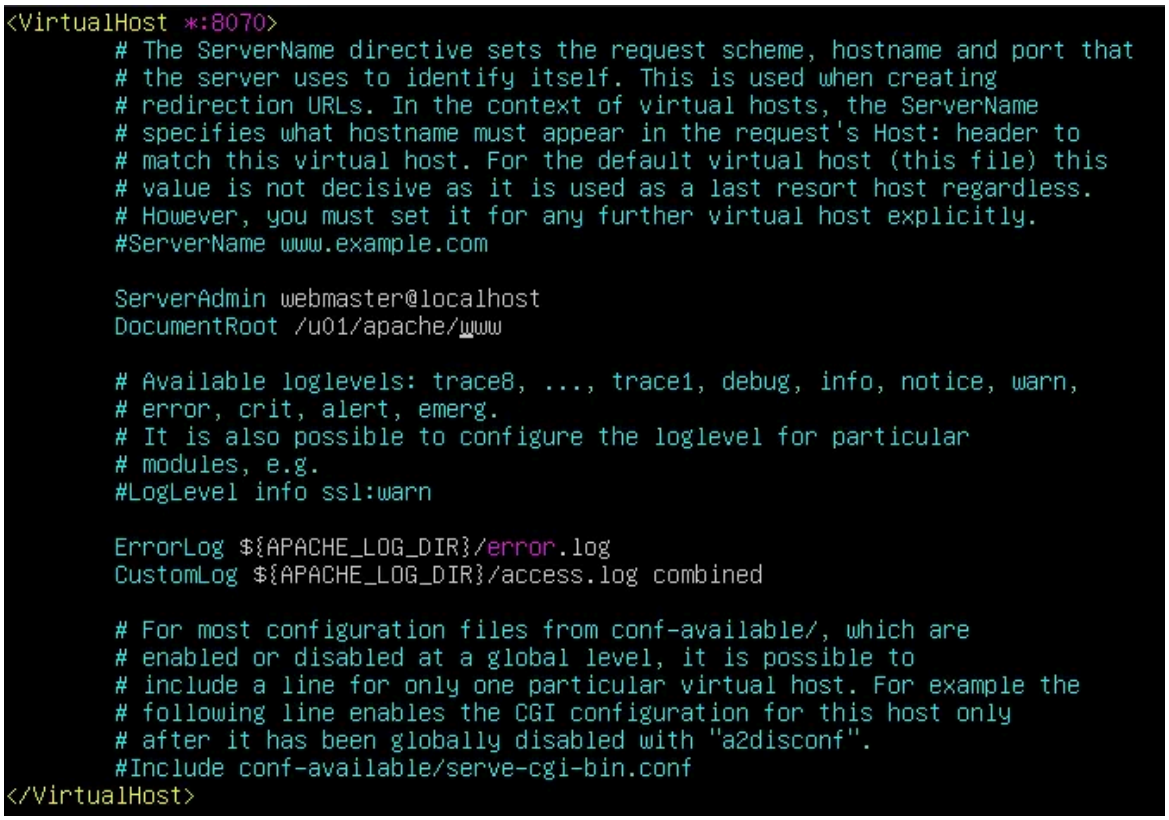


Figura 12: Configurações de site Apache.

4. Após fazer estas alterações, salve e saia do arquivo. Agora, quando reiniciarmos o Apache e tentarmos acessar “localhost:8070”, não teremos permissão, pois não temos direito de acesso a pasta “/u01”, por padrão, a pasta root, não permite acesso a nenhuma outra. Para conseguirmos acessar, teremos que adicionar permissões para que o Apache consiga acessar a pasta “/u01”. Para fazer isto, devemos acessar o arquivo “apache2.conf” no diretório “/etc/apache2/”, navegando o arquivo de configurações, devemos encontrar a seção de *directories*. Estas configurações em formato de blocos XML *(Extensible Markup Language)*, representam o nível de acesso a cada diretório de nosso sistema. Para liberar acesso ao Apache para o diretório “/u01/apache/www”, é necessário adicionar a seguinte configuração, assim como mostra na Figura 13:

```
<Directory /u01/apache/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

Figura 13: Permissões de diretórios do Apache.

5. Salve e saia do arquivo. Agora quando reiniciarmos o apache, se tudo foi feito corretamente, o site deve abrir normalmente. Para reiniciar o Apache, use:

sudo systemctl restart apache2: Reinicia o processo do Apache.

6. Agora, a partir da máquina virtual, conseguimos acessar o endereço “localhost:8070”, porém, não conseguimos acessá-lo fora da máquina virtual, em nosso sistema host. Para conseguirmos fazer isso é necessário configurar as portas necessárias nas configurações da ferramenta de virtualização. Primeiro, se desliga a máquina virtual e em seguida, clica-se com o botão direito do mouse sobre a máquina virtual criada, e segue-se o caminho “Settings > Network > Port Forwarding”, depois clica-se para adicionar uma nova regra, como mostrado na Figura 14:

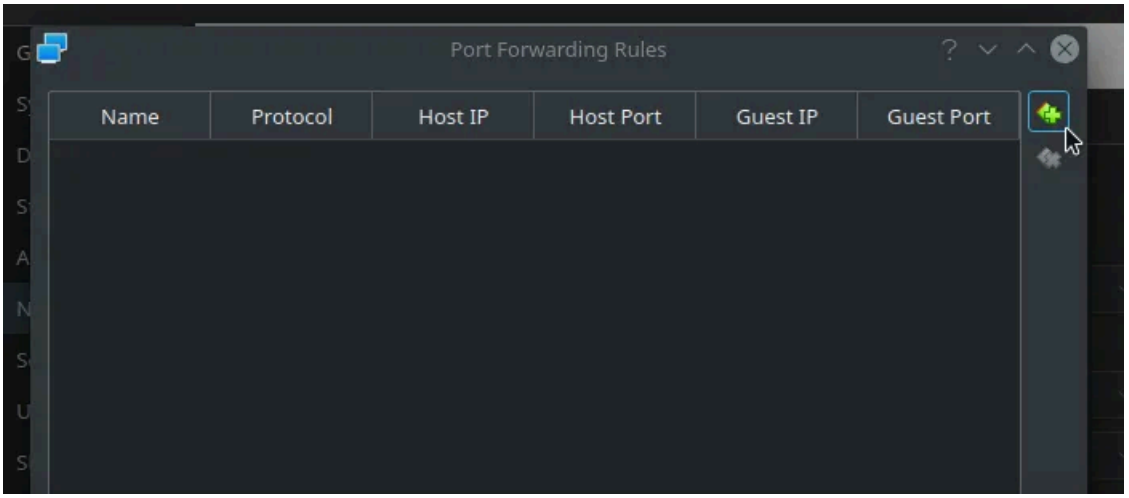


Figura 14: Página de configuração de portas no VirtualBox.

7. Agora basta editar a nova regra com a porta que se quer usar para acessar sua máquina *host* e qual a porta da máquina virtual servirá para o tunelamento para a porta do *host*, configuração representada na Figura 15:

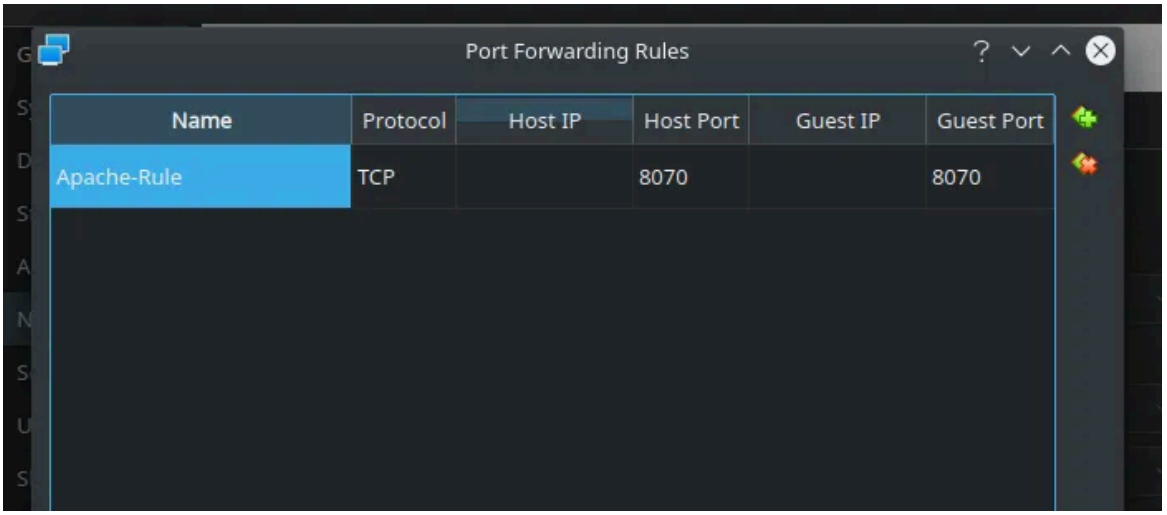


Figura 15: Configurações de portas da máquina-01 VirtualBox.

8. Esta configuração define que o *localhost* do *host* na porta 8070, refletirá o conteúdo da porta 8070 da máquina virtual. Agora marca-se OK em tudo e liga-se a máquina novamente. Para podermos acessar localhost:8070 no navegador de nossa máquina física, resta configurar a porta 8070 no *Firewall* da máquina virtual, para assim podermos ter acesso. Para permitir o

acesso externo na porta 8070 via TCP, devemos permitir este acesso no *Firewall* de nossa máquina virtual:

sudo ufw allow 8070/tcp: Permite acesso a porta 8070 via TCP.

9. Agora rodamos os seguintes comandos para reiniciar o *Firewall* e o Apache:

sudo ufw disable: Desabilita o *Firewall*.

sudo ufw enable: Inicia o *Firewall*.

sudo systemctl restart apache2: Reinicia o Apache.

10. Com isso, temos acesso total ao nosso site e arquivos no diretório "/u01/apache/www", tanto quando para acessarmos "localhost:8070" pela máquina virtual ou pelo navegador na máquina física. Desta forma podemos dar como encerrada a configuração da máquina-01.

3.1.2 – Configuração máquina-02

- **Instalação e configuração do NGNIX**

1. Após configuração da máquina-01, podemos dar início a configuração da máquina-02, para isso precisamos iniciar pela atualização de dependências do sistema, instalação do NGINX, e ativação, caso necessário:

sudo apt-get update && sudo apt-get install nginx: Atualização de dependências e instalação NGINX.

sudo systemctl status nginx: Verificar instalação.

sudo systemctl start nginx: Ativar caso o status for *"inactive"*.

2. Após a inicialização do servidor NGINX é preciso realizar a configuração do *Firewall*, assim inicializando-o e habilitando o NGINX no firewall, para validar seu status:

sudo ufw enable: Habilitar *Firewall*.

sudo ufw allow 'Nginx HTTP': Habilitar NGINX no *Firewall*.

sudo ufw status: Verificar Status.

3. O retorno esperado é NGINX HTTP como "*Allow*" no modo (*Anywhere* (v6)). Na próxima etapa é necessário realizar a configuração da porta, para que a máquina física consiga ter acesso ao NGINX que está rodando na máquina virtual, para isso é necessário adicionar a seguinte regra no *firewall*:

sudo ufw allow 8090/tcp: Libera a porta 8090 no *firewall* via protocolo TCP.

4. Agora devemos alterar a execução do NGINX, que está operando na porta 80, e apontar para a porta 8090. Para isto, basta modificar a porta no arquivo de configurações "*default*" localizado em "/etc/nginx/sites-available/default", basta realizar a alteração contida na Figura 16:

```
...
#Default server configuration
#
server{
        listen 8090 default_server;
        listen [::]:8090 default_server;
...
```

Figura 16: Configuração de portas do NGINX.

5. Por fim, para ter acesso externo na máquina-02, é necessário fazer uma configuração semelhante à configuração de redes do VirtualBox realizada anteriormente na máquina-01. Desliga-se a máquina-02 e em seguida, clica com o botão direito do mouse na máquina-02 e segue-se o caminho "Settings > Network > Port Forwarding", depois clica-se para adicionar uma nova regra, e se adiciona o apontamento para a porta 8090, exemplo na Figura 17:

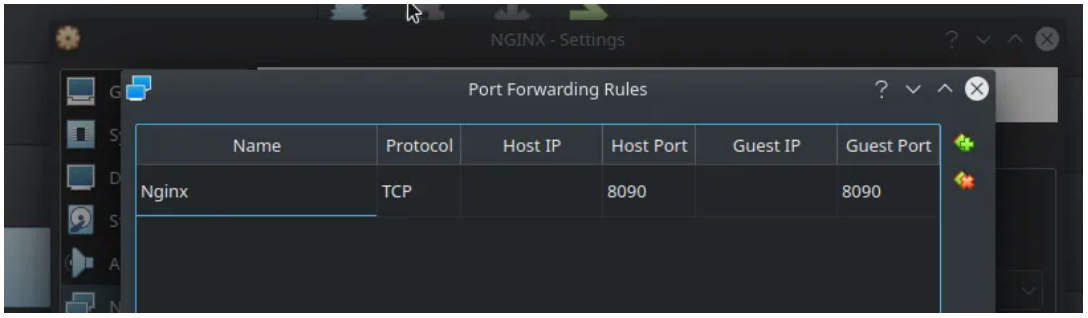


Figura 17: Configuração de portas VirtualBox na máquina-02.

Agora, ao acessar localhost na porta 8090, deve conseguir acessar a máquina-01. Com isto funcionando, podemos seguir para o próximo passo e adicionar o arquivo “index.html” na máquina-02. Basta adicionar no diretório “/var/www/html/index.html”, assim se dá como encerrada a configuração da máquina-02.

3.1.3 – Testando as máquinas

Com ambas as máquinas configuradas, ao chamar “localhost:<porta_maquina(8090, 8070)>” as máquinas devem conseguir requisitar-se entre si e redirecionarem uma à outra. Ao acessar a máquina-02, sendo a máquina de acesso, via localhost:8090, conseguimos conectar ao NGINX, representado na Figura 18:

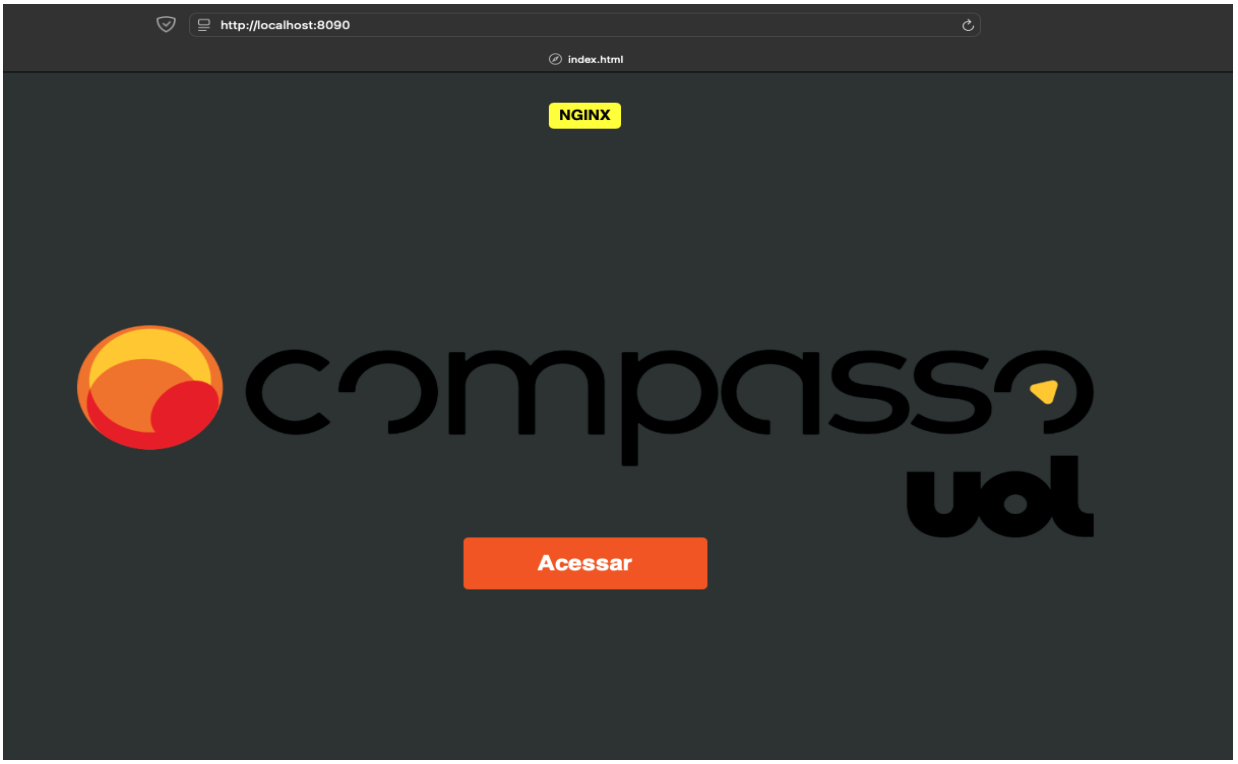


Figura 18: Resultado máquina-02 via NGINX.

Ao clicar em "Acessar" na página anterior da máquina-02, seremos redirecionados para a página da máquina-01, em localhost:8070, via Apache, como visto na Figura 19:

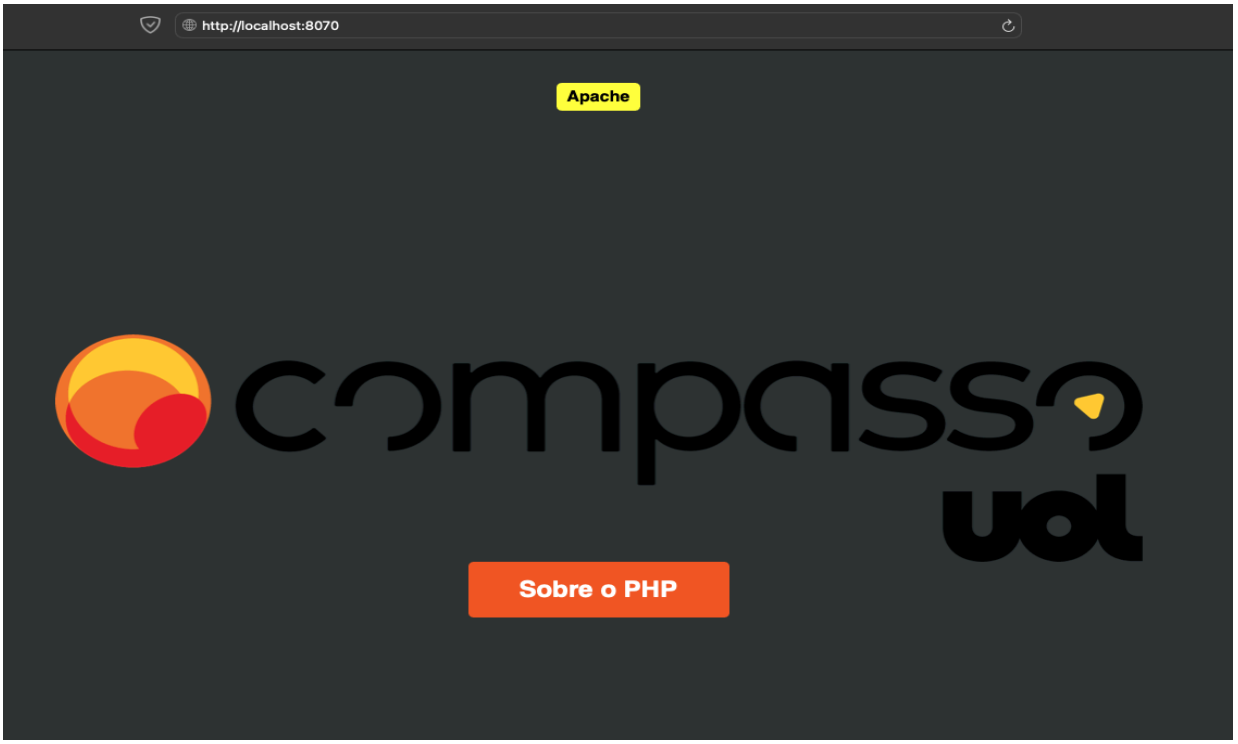
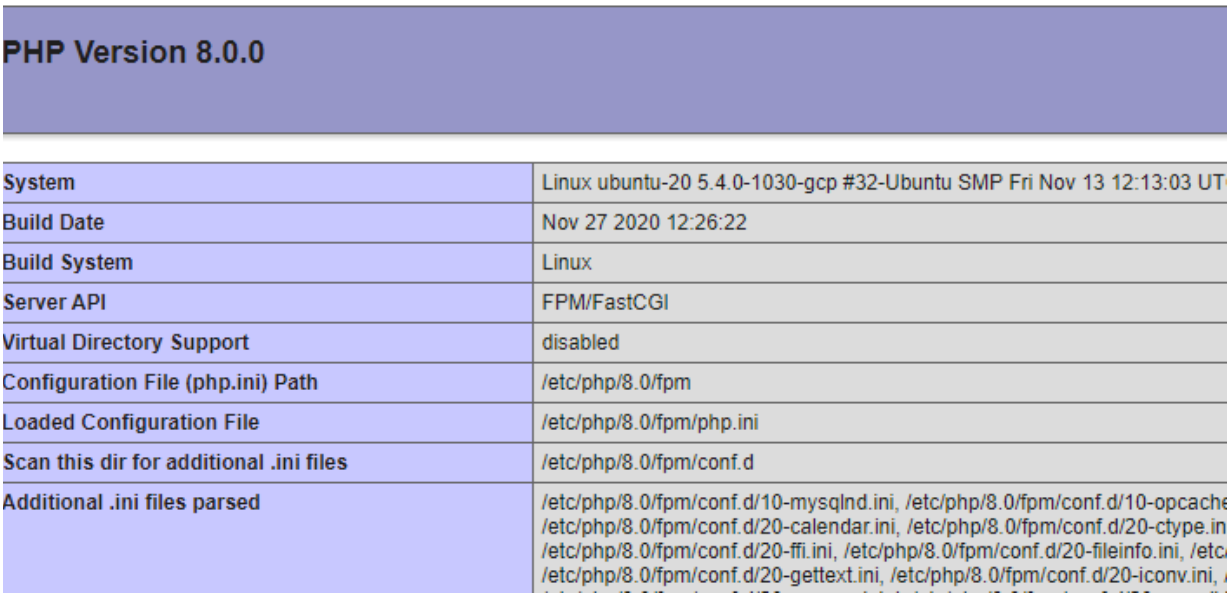


Figura 19: Resultado máquina-01 via Apache.

Ao clicar em "Sobre o PHP", na máquina-01, Figura 19, mantemos na porta 8070, e conseguimos ver as informações do PHP da máquina-01, como mostrado na Figura 20:



PHP Version 8.0.0

System	Linux ubuntu-20 5.4.0-1030-gcp #32-Ubuntu SMP Fri Nov 13 12:13:03 UT
Build Date	Nov 27 2020 12:26:22
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.0/fpm
Loaded Configuration File	/etc/php/8.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.0/fpm/conf.d
Additional .ini files parsed	/etc/php/8.0/fpm/conf.d/10-mysqlnd.ini, /etc/php/8.0/fpm/conf.d/10-opcache /etc/php/8.0/fpm/conf.d/20-calendar.ini, /etc/php/8.0/fpm/conf.d/20-ctype.in /etc/php/8.0/fpm/conf.d/20-ffi.ini, /etc/php/8.0/fpm/conf.d/20-fileinfo.ini, /etc /etc/php/8.0/fpm/conf.d/20-gettext.ini, /etc/php/8.0/fpm/conf.d/20-iconv.ini,

Figura 20: Resposta php info máquina-01.

3.2 – Segunda Lista de cursos

No fim dos exercícios Linux, seguimos com uma nova lista de cursos para consumirmos, conforme o Quadro 2 demonstra, abaixo:

Quadro 2 - Segunda lista de cursos.

CURSO	TEMA	HORAS
Redes	Redes parte 1: Introdução, Conceitos e Prática	4h
Redes	DNS: Entenda a resolução de nomes na internet	8h
Docker	Docker: Criando containers sem dor de cabeça	8h
Kubernetes	Kubernetes: Pods, Services e ConfigMaps	8h
Kubernetes	Kubernetes: Deployments, Volumes e Escalabilidade	8h

- **Redes parte 1: Introdução, Conceitos e Prática:** Neste módulo, foi feita uma introdução ao terminal Linux, uma ferramenta essencial para usar o sistema. Foram ensinados comandos básicos para navegar pelos arquivos, como ‘ls’ para listar diretórios e ‘cd’ para mudar de pasta. Além disso, foram abordadas as operações de criação, remoção e edição de arquivos, formando uma base sólida para o uso do Linux.

- **Redes: DNS: Entenda a resolução de nomes na internet:** Este módulo explicou como funciona o Domain Name System (DNS) e sua importância na navegação pela internet. Foram abordados conceitos como a resolução de nomes e os tipos de registros DNS, além da relação entre endereços IP e nomes de domínio.

- **Docker: Criando containers sem dor de cabeça:** Neste módulo, foram ensinados os conceitos de como funcionam os containers utilizando Docker. O foco foi a criação e gerenciamento de containers, mostrando como facilitar a implantação de aplicações. Em exercícios práticos, executamos a construção de imagens e o uso de comandos que ajudam a trabalhar com containers de forma mais eficiente.

- **Kubernetes: Pods, Services e ConfigMaps:** Este módulo apresentou os principais componentes do Kubernetes, como Pods, Services e ConfigMaps. Foram ensinados conceitos de orquestração de containers, permitindo entender como gerenciar aplicações distribuídas. A prática deste módulo foi criar esses elementos, mostrando em situação real como escalar e manter aplicações de forma confiável.

- **Kubernetes: Deployments, Volumes e Escalabilidade:** Aqui, foram discutidos Deployments e Volumes, com foco na gestão de aplicações de grande escala. O módulo ensinou como implementar estratégias de escalabilidade e alta disponibilidade, que são fundamentais para garantir que as aplicações consigam atendam à demanda requisitada. O aprendizado sobre o uso de volumes para persistência de dados, mostra o quanto é importante seu uso em alguns casos, evitando manutenção individual dos dados de cada instância da aplicação e garantindo que estes não sejam perdidos em caso de indisponibilidade da aplicação.

3.2.1 – Exercícios de fixação Kubernetes

Após a segunda trilha de cursos, foi aplicado um novo desafio como verificação de aprendizagem dos cursos, onde havia duas atividades: criar NGINX e criar Wordpress

- **Criar NGINX**

Na primeira atividade, seria criar um namespace e dois pods Nginx, configurando um Load Balancer para balancear a carga entre os dois.

1. Primeiro, crie um namespace chamado labnginx a partir do seguinte comando:

kubectl create namespace labnginx: Cria um namespace no Kubernetes.

2. Após criar o namespace, é necessário criar um arquivo nginx-deployment.yaml, onde haverá as configurações para criar dois pods nginx. neste namespace, dentro de um único

ingress, onde o Load balancer deverá distribuir a carga entre estes dois pods. Crie e adicione no arquivo, assim como na Figura 21, 22 e 23:

```yaml
### nginx-deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: labnginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80

---
```

Figura 21: Arquivo IAC nginx-deployment.yaml.

```yaml
---

apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: labnginx
spec:
  type: ClusterIP
  selector:
    app: nginx
  ports:
  - port: 80
    targetPort: 80

---
```

Figura 22: Continuação 2 arquivo IAC nginx-deployment.yaml.

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
  namespace: labnginx
spec:
  rules:
  - host: nginx.local
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: nginx-service
            port:
              number: 80
```

Figura 23: Continuação 3 arquivo IAC nginx-deployment.yaml.

3. Após a criação do arquivo, precisamos aplicá-lo no kubernetes local, para isso executa-se o seguinte comando:

kubectl apply -f nginx-deployment.yaml: Aplica o arquivo no kubernetes.

4. Ao abrir o ip das máquinas nas suas respectivas portas deve retornar uma tela semelhante a da Figura 24 no navegador:

Figura 24: Retorno NGINX kubernetes.

5. Para validar o funcionamento desta parte da atividade, as chamadas deveriam ser balanceadas entre os dois pods, para isto, basta abrir os logs de cada máquina e verificar que a cada vez que é feita uma chamada, cada chamada irá requisitar em uma máquina diferente, com isto, a atividade está encerrada.

- **Criar Wordpress**

Na segunda atividade foi necessário montar um ambiente para um WordPress, que deveria ter um banco de dados MySQL, incluindo ajustes na porta e na persistência de dados. Além disso, configurar o Wordpress com seu próprio armazenamento persistente.

1. Este ambiente é separado do anterior, então é necessário criar um namespace separado, para isso seguimos criando o namespace labwordpress:

kubectl create namespace labwordpress: Cria um namespace no Kubernetes.

2. Para criação do serviço MySQL, foi criado um arquivo “mysql-service.yaml”, para externalizar a porta padrão para “3308”, apontando para a interna “3306”, como na Figura 25:

```yaml
# mysql-service.yaml

apiVersion: v1
kind: Service
metadata:
  name: mysql-service
  namespace: labwordpress
spec:
  type: ClusterIP
  ports:
  - port: 3308
    targetPort: 3306
  selector:
    app: mysql
```

Figura 25: Service MySQL.

3. Depois foi criado um arquivo secret para armazenar a senha do MySQL, garantindo segurança através da codificação em base64. A senha utilizada foi “UzNuSDRGMHJUZQo=” O arquivo “mysql-secret.yaml” aplicado é correspondente a Figura 26:

```yaml
# mysql-secret.yaml

apiVersion: v1
kind: Secret
metadata:
  name: mysql-secret
  namespace: labwordpress
type: Opaque
data:
  mysql-password: UzNuSDRGMHJUZQo=
```

Figura 26: Arquivo secret do kubernetes para o MySQL.

4. Após criar a secret para o MySQL, criamos um “*Persistent Volume Claim* (PVC)” de 3GB de capacidade para o MySQL, onde seriam persistidos os dados, caso os pods parassem de funcionar por qualquer razão. O arquivo “mysql-pvc.yaml” aplicado, assim como a Figura 27:

```yaml
# mysql-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pvc
  namespace: labwordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

Figura 27: PVC kubernetes para o MySQL.

5. Então foi possível criar o "*deployment*" que irá executar nosso MySQL, e direcionar nossa persistência como "*volume mount*" e também referenciar as *secrets* criadas anteriormente.

Este foi o "mysql-deployment.yaml" criado, conforme a Figura 28:

```yaml
# mysql-deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql-deployment
  namespace: labwordpress
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysql
        image: mysql:5.7
        ports:
        - containerPort: 3306
        env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-password
        volumeMounts:
        - name: mysql-persistent-storage-lab
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-persistent-storage-lab
        persistentVolumeClaim:
          claimName: mysql-pvc
```

Figura 28: Deployment do MySQL.

6. Após criação do MySQL, prosseguimos para a criação do Wordpress, primeiro começamos com o service, “wordpress-service.yaml” configurado com a porta “80”, código presente na Figura 29:

```yaml
# wordpress-service.yaml

apiVersion: v1
kind: Service
metadata:
  name: wordpress-service
  namespace: labwordpress
spec:
  type: ClusterIP
  ports:
  - port: 80
    targetPort: 80
  selector:
    app: wordpress
```

Figura 29: Service do wordpress.

7. O Wordpress também precisa de um PVC de 3GB, para caso algum pod tenha indisponibilidade, o site Wordpress e suas configurações irão manter-se salvos, “wordpress-pvc.yaml”. O código respectivo a isso pode ser visto na Figura 30:

```yaml
# wordpress-pvc.yaml

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wordpress-pvc
  namespace: labwordpress
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

Figura 30: PVC do Wordpress.

8. Por fim, aplicamos o *deployment*, apontando para o MySQL e para o seu respectivo PVC, basta aplicar o arquivo “wordpress-deployment.yaml”, conforme a Figura 31:

```yaml
# wordpress-deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-deployment
  namespace: labwordpress
spec:
  replicas: 1
  selector:
    matchLabels:
      app: wordpress
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
      - name: wordpress
        image: wordpress:latest
        ports:
        - containerPort: 80
        env:
        - name: WORDPRESS_DB_HOST
          value: "mysql-service:3308"
        - name: WORDPRESS_DB_USER
          value: "root"
        - name: WORDPRESS_DB_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-password
        volumeMounts:
        - name: wordpress-persistent-storage-lab
          mountPath: /var/www/html
      volumes:
      - name: wordpress-persistent-storage-lab
        persistentVolumeClaim:
          claimName: wordpress-pvc
```

Figura 31: *Deployment* do Wordpress.

9. Para confirmar que tudo foi instalado corretamente, ao acessar o IP do pod do Wordpress, deve conseguir executar o serviço e retornar algo semelhante ao que é mostrado na Figura 32:

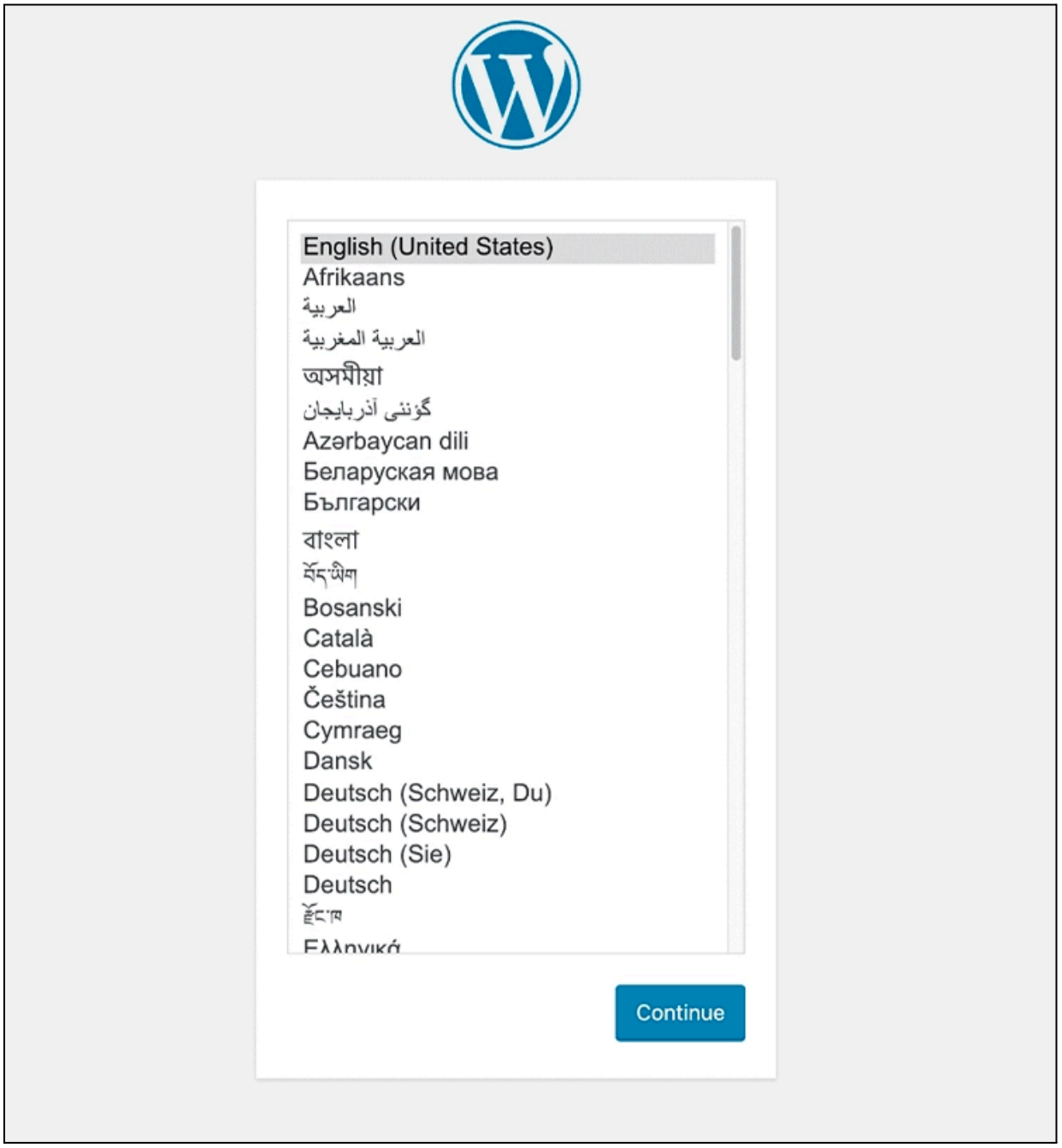


Figura 32: Tela inicial de instalação do Wordpress.

3.3 – Terceira Lista de Cursos

Quadro 3 - Terceira Lista de Cursos

CURSO	TEMA	HORAS
AWS: Conhecimentos Gerais	Deploy no Amazon EC2: Alta disponibilidade de uma aplicação	10h
AWS: Conhecimentos Gerais	Amazon Lightsail: Descomplicando a nuvem	8h
AWS: Conhecimentos Gerais	Amazon CloudWatch: Visibilidade completa das aplicações e serviços na nuvem	6h
AWS: Conhecimentos Gerais	Amazon S3: Manipule e armazene objetos na nuvem	12h
AWS: Conhecimentos Gerais	Amazon VPC: Provisione uma nuvem privada	6h
AWS: Conhecimentos Gerais	Amazon CloudFront e Route53: Distribua os arquivos e crie seu próprio domínio	8h
AWS: Conhecimentos Gerais	Amazon ECS: Gerencie Docker na nuvem da AWS	8h
AWS: Conhecimentos Gerais	Amazon Elastic Beanstalk Parte 1: Containers Docker	6h
AWS: Conhecimentos Gerais	Amazon Elastic Beanstalk Parte 2: Containers e NGINX	8h
AWS Certified Cloud Practitioner	AWS Certified Cloud Practitioner: Domain 1 e 2	8h
AWS Certified Cloud Practitioner	AWS Certified Cloud Practitioner: Domain 3 e 4	10h

- **Deploy no Amazon EC2: Alta disponibilidade de uma aplicação:** Neste módulo, foram abordados os princípios de alta disponibilidade usando o Amazon EC2. Foi ensinado como configurar instâncias de forma a garantir que uma aplicação permaneça acessível, mesmo em caso de falhas.

- **Amazon Lightsail: Descomplicando a nuvem:** Este módulo apresentou o Amazon Lightsail como uma forma simplificada de usar a nuvem. Foram discutidos recursos e ferramentas que facilitam o lançamento de aplicações de forma descomplicada, o que mostrou como de iniciar projetos na nuvem de maneira acessível.

- **Amazon CloudWatch: Visibilidade completa das aplicações e serviços na nuvem:** Neste módulo, foi ensinado a monitorar aplicações e serviços utilizando o Amazon CloudWatch. O foco foi na coleta de métricas e na criação de alarmes, permitindo uma visão clara do desempenho das aplicações. Uma habilidade crucial para sempre verificar a saúde dos sistemas em nuvem, e ajudar para que tenhamos uma rápida reação em casos de falhas.

- **Amazon S3: Manipule e armazene objetos na nuvem:** Aqui, foram explorados os fundamentos do Amazon S3, uma solução para armazenamento de objetos. O módulo ensinou como armazenar, recuperar e gerenciar dados na nuvem, incluindo boas práticas para garantir a segurança e a eficiência no uso do espaço.

- **Amazon VPC: Provisione uma nuvem privada:** Neste módulo, foram abordados os conceitos de Virtual Private Cloud (VPC). Foi ensinado como criar uma rede privada na AWS, incluindo a configuração de subnets e regras de segurança. Essa compreensão é essencial para garantir a proteção dos dados em um ambiente de nuvem.

- **Amazon CloudFront e Route53: Distribua os arquivos e crie seu próprio domínio:** Este módulo apresentou o Amazon CloudFront e o Route 53, focando na distribuição de conteúdo e gerenciamento de domínios. Mostrou como configurar a entrega rápida de arquivos e a criação de um domínio personalizado, melhorando a experiência do usuário.

- **Amazon ECS: Gerencie Docker na nuvem da AWS:** Neste módulo, foram ensinados os conceitos de Amazon ECS para gerenciamento de containers Docker na AWS. O foco foi na implantação e na escalabilidade de aplicações baseadas em containers, facilitando o uso do Docker na nuvem.

- **Amazon Elastic Beanstalk Parte 1: Containers Docker**: Aqui, o curso abordou o Amazon Elastic Beanstalk e sua utilização com containers Docker. Aprender a implementar aplicações em um ambiente gerenciado simplifica a entrega e a operação de software, o que permite que os desenvolvedores se concentrem no código.

- **Amazon Elastic Beanstalk Parte 2: Containers e NGINX:** Neste módulo, a continuação do estudo do Elastic Beanstalk envolveu a configuração de NGINX para gerenciar aplicações. Aprendeu-se a integrar esse servidor web com containers, aprimorando a capacidade de entrega de aplicações.

- **AWS Certified Cloud Practitioner: Domain 1 e 2:** Este módulo abordou os fundamentos da AWS, incluindo conceitos essenciais sobre a nuvem e a infraestrutura da AWS. O conteúdo foi projetado para preparar para a certificação, cobrindo temas fundamentais que são a base do conhecimento em nuvem e temas da prova para a certificação AWS Certified Cloud Practitioner.

- **AWS Certified Cloud Practitioner: Domain 3 e 4:** Aqui, foram discutidos aspectos mais avançados da AWS, em continuação ao módulo anterior, incluindo questões de segurança e conformidade, além de estratégias de implementação e gerenciamento de custos. Esses conhecimentos são importantes para um entendimento completo das melhores práticas em nuvem.

3.4 – Quarta Lista de Cursos

Quadro 4 - Quarta lista de cursos.

CURSO	**TEMA**	**HORAS**
Conceitos DevSecOps	The Twelve-Factor App: Metodologia para construção de aplicações robustas	10h
Conceitos DevSecOps	Git e Github: Controle e compartilhe seu código	6h
Conceitos DevSecOps	Git e Github: Estratégias de ramificação, Conflitos e Pull Requests	8h
Conceitos DevSecOps	Integração Contínua: Mais qualidades e menos risco no desenvolvimento	6h
Conceitos DevSecOps	Entrega Contínua: Confiabilidade e qualidade na implantação de software	8h
Conceitos DevSecOps	Jenkins e Docker: Pipeline de entrega contínua	12h
Conceitos DevSecOps	Jenkins: Integração contínua	8h
Conceitos DevSecOps	Gitlab CI e Docker: Pipeline de entrega contínua	8h
Conceitos DevSecOps	Maven: Gerenciamento de dependências e build de aplicações Java	8h
Conceitos DevSecOps	Build de aplicação .NET: construindo seus projetos com MSBuild	10h
Conceitos DevSecOps	Ansible: Sua infraestrutura como código	12h
Conceitos DevSecOps	Terraform: Automatize a infraestrutura na nuvem	8h
Conceitos DevSecOps	Grafana e Telegraf: Monitorando em tempo real	10h

- **The Twelve-Factor App: Metodologia para construção de aplicações robustas:** Neste módulo, foram apresentados os princípios da metodologia Twelve-Factor App, que visa construir aplicações mais escaláveis e robustas. Os fatores que ajudam a garantir que as aplicações sejam fáceis de manter e de implantar, promovendo uma abordagem estruturada para o desenvolvimento.

- **Git e Github: Controle e compartilhe seu código:** Este módulo abordou o uso do Git e do GitHub para controle de versão. Foram ensinadas as práticas de *commit, push* e *pull*, permitindo uma melhor organização do código e facilitando a colaboração entre equipes. O aprendizado sobre o gerenciamento de versões é fundamental para qualquer projeto de software.

- **Módulo: Git e Github: Estratégias de ramificação, Conflitos e Pull Requests:** Neste módulo, foram exploradas estratégias de ramificação, como criar *branches* e lidar com conflitos no Git ao fazer Merges. O conceito de Pull Requests foi abordado, mostrando como revisar e integrar alterações de forma colaborativa. Essas são habilidades essenciais para manter um fluxo de trabalho eficiente nas equipes de desenvolvimento.

- **Integração Contínua: Mais qualidades e menos risco no desenvolvimento:** Aqui, foram discutidos os conceitos de Integração Contínua (CI), que visam detectar problemas rapidamente durante o desenvolvimento. O módulo ensinou como automatizar testes e verificações, garantindo que as mudanças no código não quebrem funcionalidades existentes.

- **Entrega Contínua: Confiabilidade e qualidade na implantação de software:** Este módulo focou na Entrega Contínua (CD) e como esta melhora a confiabilidade das implantações. Esse curso mostrou como o fato de automatizar o processo de entrega de software, pode reduzir riscos, aumentando a qualidade das versões lançadas.

- **Jenkins e Docker: Pipeline de entrega contínua:** Neste módulo, foram ensinados os conceitos de Jenkins e Docker para criar pipelines de entrega contínua. Ensinou como integrar essas ferramentas para automatizar a construção, testes e *deploy* de aplicações, aumentando a eficiência no ciclo de desenvolvimento.

- **Jenkins: Integração contínua:** Este módulo aprofundou os conhecimentos sobre Jenkins, focando na configuração de *jobs* de integração contínua. Ensinou a configurar testes automatizados e manter relatórios, garantindo que o código seja constantemente verificado.

- **Gitlab CI e Docker: Pipeline de entrega contínua:** Aqui, foram exploradas as funcionalidades do GitLab CI para automação de pipelines com Docker. Mostrando como criar e gerenciar pipelines de forma prática, facilitando a integração e entrega de aplicações.

- **Maven: Gerenciamento de dependências e build de aplicações Java:** Neste módulo, foram abordados os conceitos de Maven, uma ferramenta para gerenciar dependências e fazer *build* de aplicações Java. O aprendizado incluiu a configuração de projetos e a importância de manter as dependências organizadas.

- **Build de aplicação .NET: construindo seus projetos com MSBuild:** Este módulo ensinou o uso do MSBuild para construir projetos .NET. Foram discutidas as melhores práticas para compilar e gerenciar dependências.

- **Ansible: Sua infraestrutura como código:** Neste módulo, foi apresentado o Ansible, uma ferramenta para gerenciar infraestrutura como código. Aprendemos a automatizar a configuração de servidores e a garantir consistência na infraestrutura, facilitando a gestão de ambientes.

- **Terraform: Automatize a infraestrutura na nuvem:** Aqui, foram discutidos os conceitos do Terraform, para automatizar a criação e gestão de infraestrutura em nuvem. O módulo ensinou a usar código para provisionar recursos, promovendo a agilidade no gerenciamento de ambientes na nuvem.

- **Grafana e Telegraf: Monitorando em tempo real:** Este módulo apresentou o Grafana e o Telegraf como ferramentas de monitoramento. Mostrou como coletar métricas em tempo real e a visualizar dados, permitindo um bom acompanhamento do desempenho das aplicações e da infraestrutura.

4 – Considerações finais

Este estágio foi imprescindível para minha carreira como profissional da área de tecnologia, foi a oportunidade que me posicionou no mercado.

Durante o estágio pude ter acesso ao uso de ferramentas poderosas e complexas, e graças aos conhecimentos que já havia adquirido no curso TADS, já tinha tido contato com alguns temas, e em temas nunca vistos antes, tive alguma facilidade. No estágio aprofundei conhecimentos de temas da minha formação acadêmica, assim como desenvolvi habilidades de comunicação, programação, aprendi e experienciei metodologias ágeis, criação e configuração de sites, redes, Linux, e virtualização.

No fim do estágio fui convidado a ser internalizado na companhia, onde aceitei e atuo como DevOps até então.

Hoje com toda a bagagem do estágio, e alguns anos de trabalho na área de DevOps, vejo quão importante foram a formação do estágio, e do curso TADS. Estas me prepararam me moldando para ser um profissional comunicativo, responsável, com visão analítica para resolução de problemas, e em constante crescimento.

Expresso grande gratidão, pela oportunidade oferecida pela companhia Compass.UOL, na qual me fez aprender muito e me deu a chance de provar meu valor como profissional.

A instituição IF Goiano, sou grato porque me colocou em contato com excelentes professores nos quais não só me ensinaram, mas me auxiliaram a moldar minha melhor versão, como profissional, e principalmente como pessoa. E também aos colegas, dos quais muitos se tornaram grandes amigos.

Referências

AGÊNCIA BRASIL. ***Home office foi adotado por 46% das empresas durante pandemia.*** Disponível em: https://agenciabrasil.ebc.com.br/economia/noticia/2020-07/home-office-foi-adotado-por-46-das-empresas-durante-pandemia. Acesso em: 25 ago. 2024.

AWS. ***What is AWS?*** Disponível em: https://aws.amazon.com/pt/what-is-aws/. Acesso em: 02 nov. 2024.

AWS. ***What is DevSecOps?*** Disponível em: https://aws.amazon.com/what-is/devsecops/. Acesso em: 26 ago. 2024.

AWS. ***What is Digital Transformation?*** Disponível em: https://aws.amazon.com/pt/what-is/digital-transformation/. Acesso em: 25 ago. 2024.

ATLASIAN. ***Agile Software Development with Atlassian.*** Disponível em: https://www.atlassian.com/br/agile/scrum. Acesso em: 26 ago. 2024.

ATLASIAN. ***Sprints – Scrum at Atlassian.*** Disponível em: https://www.atlassian.com/br/agile/scrum/sprints. Acesso em: 25 ago. 2024.

BIZNEWS. ***UOL Diveo se reposiciona no mercado, agora é Compasso.*** Disponível em: https://www.biznews.com.br/uol-diveo-se-reposiciona-no-mercado-agora-e-compasso/. Acesso em: 25 ago. 2024.

COMPASS. ***Compasso 100% para casa em breve.*** Disponível em: https://blog.compass.uol/noticias/compasso-100-para-casa-em-breve/. Acesso em: 25 ago. 2024.

COMPASS. ***Em reestruturação, UOL Diveo separa áreas de negócio e muda nome.*** Disponível em: https://blog.compass.uol/noticias/em-reestruturacao-uol-diveo-separa-areas-de-negocio-e-muda-nome/. Acesso em: 25 ago. 2024.

COMPASS. ***Home Office: Como a tecnologia impulsionou um novo paradigma de trabalho.*** Disponível em: https://blog.compass.uol/noticias/home-office-como-a-tecnologia-impulsionou-um-novo-paradigma-de-trabalho/. Acesso em: 25 ago. 2024.

COMPASS. ***Política de Gestão de Qualidade.*** Disponível em: https://compass.uol/pt/politica-de-gestao-de-qualidade. Acesso em: 25 ago. 2024.

DESCOMPLICANDO DOCKER. ***Descomplicando Docker – Livro online.*** Disponível em: https://livro.descomplicandodocker.com.br. Acesso em: 30 out. 2024.

GOOGLE CLOUD. ***What is Kubernetes?*** Disponível em: https://cloud.google.com/learn/what-is-kubernetes?hl=pt-BR. Acesso em: 30 out. 2024.

IBM. ***What is Kubernetes?*** Disponível em: https://www.ibm.com/br-pt/topics/kubernetes. Acesso em: 30 out. 2024.

LINUX ESSENTIALS. ***What is Linux*.** Bresnahan, Christine; Blum, Richard. [S.l.]: Christine Bresnahan & Richard Blum, 2020. Disponível em: https://fedoraproject.org/wiki/Yum. Acesso em: 30 out. 2024.

POULTON, Nigel. ***The Kubernetes Book: 2024 Edition*.** 2021. E-book (235p.). Disponível em: https://leanpub.com/thekubernetesbook. Acesso em: 30 out. 2024.

RED HAT. ***What is CI/CD?*** Disponível em: https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd. Acesso em: 26 ago. 2024.

RED HAT. ***What is Docker?*** Disponível em: https://www.redhat.com/pt-br/topics/containers/what-is-docker. Acesso em: 30 out. 2024.

SCHWABER, Ken; SUTHERLAND, Jeff. ***O Guia do Scrum - Recurso eletrônico*. LIVRO: O guia definitivo para o Scrum: as regras do jogo.** [S.l.]: Ken Schwaber and Jeff Sutherland, 2020. Disponível em: https://pergamum-biblioteca.pucpr.br/acervo/359418. Acesso em: 26 ago. 2024.

SUTHERLAND, J. ***Scrum, a arte de fazer o dobro do trabalho em metade do tempo*.** [S.l.]: Leya, 2014.

UNASUS. ***Organização Mundial de Saúde declara pandemia de coronavírus*.** Disponível em: https://www.unasus.gov.br/noticia/organizacao-mundial-de-saude-declara-pandemia-de-coronavirus. Acesso em: 25 ago. 2024.

VIVA SECURITY. ***Shift Left Security in Cybersecurity*.** Disponível em: https://blog.vivasecurity.com.br/ciberseguranca/shift-left-security/. Acesso em: 26 ago. 2024.