**Documentação Avaliação DevOps Jr.**

As informações iniciais sobre versões e alguns passos foram pegas no site de documentação do OpenCMS no link:

|  |
| --- |
| https://documentation.opencms.org/opencms-documentation/introduction/installation/ |

Para iniciar o laboratório foi escolhido o Debian que foi um Linux que utilizei durante a faculdade. A princípio iniciei os testes com o Debian 12, porém após vários erros de compatibilidade fiz a troca para o Debian 11, que tinha todos os pacotes que eram necessários. Foi criado a *VM (Virtual Machine)* com as informações padrão de criação e utilizando o *Graphical install* para as configurações finais.

* Após a instalação e finalização do Debian 11 é necessário logar no usurário root e utilizar o comando:

|  |
| --- |
| apt update |

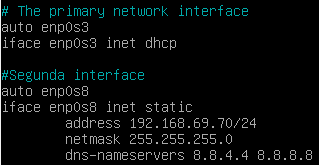
Prestado para verificar se tem alguma atualização que precisa ser feita, e, caso tenha que atualizar, utilizar o comando:

|  |
| --- |
| apt upgrade -y |

Sendo o “-y” para já aceitar qualquer pergunta feita pelo instalador. Esse comando instala as novas atualizações e isso garante que todos os pacotes estejam atualizados antes de instalar qualquer outra aplicação.

* A minha VM está configurada com 2 interfaces de rede, sendo a primeira em modo NAT e a segunda HostOnly. Nesse caso precisei fazer a configuração manual da segunda interface para que funcione o *IP (Internet Protocol)* da máquina para fazer os testes na máquina hospedeira. Para editar esse arquivo de configuração é necessário utilizar o comando:

|  |
| --- |
| nano /etc/network/interfaces |

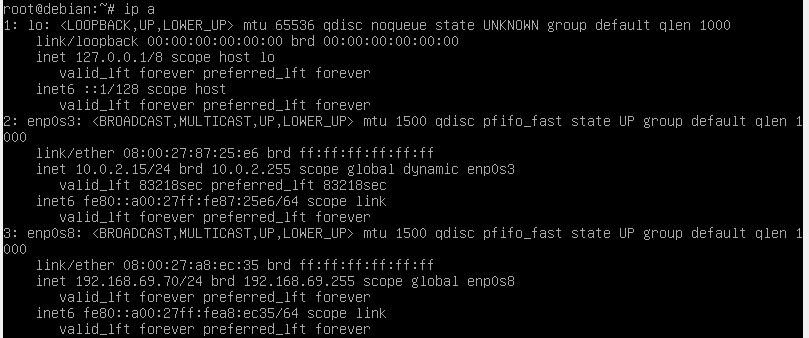


Com o arquivo configurado com essas informações, as interfaces puderam funcionar localmente. Ao configurar o IP estático com sua máscara. Após salvar será exigido reiniciar o serviço de rede com o comando:

|  |
| --- |
| systemctl restart networking |

E para testar se funcionou é usado dois comandos e suas respectivas respostas:

|  |
| --- |
| ip a |



|  |
| --- |
| ip route |



Com esses comandos é possivel ver que a máquina utiliza como *default* a interface *NAT (Network Address Translation)* para acesso a internet e configurada com o IP “192.168.69.70” pelo HostOnly. É possivel conferir se há internet com um simples “ping -c 4 8.8.8.8”, por exemplo.

* Em seguida é necessário instalar o Java (Openjdk 11) com o comando:

|  |
| --- |
| apt install -y openjdk-11-jdk |

E, logo após, é possível conferir se a versão foi a correta com o comando:

|  |
| --- |
| java -version |

O Java 11, nesse caso, irá servir como plataforma para executar as aplicações Java que irão ser utilizadas mais a frente como o Tomcate o OpenCMS. A versão 11 foi a mostrada na documentação já citada, por ser estável, segura e compatível com as outras aplicações. Que será o Tomcat 9e o OpenCMS 13.

Ainda sobre os comandos, o “apt” é utilizado para o gerenciamento de pacotes, sendo capaz de instalar, atualizar, remover e buscar pacotes dentro do Debian.

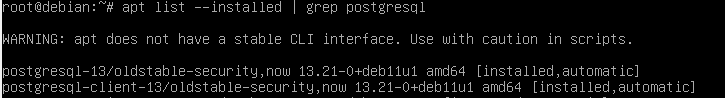
* Seguidamente realizo a instalação do banco de dados PostgreSQL com o comando:

|  |
| --- |
| apt install -y postgresql postgresql-contrib |

O comando “apt install -y postgresql” instala o servidor PostgreSQL e o comando “postgresql-contrib” serve para instalar ferramentas e extensões adicionais que possam ser úteis.

Para ver a versão do *PostgresSQL* instalado é preciso usar o comando:

|  |
| --- |
| apt list --installed | grep postgresql |



O comando “apt list –installed” mostra todos os pacotes instalados no Debian via apt, e adicionado o “grep postgresql”, ele filtra esses pacotes e mostra somente aqueles que tem postgresql no nome. Após dar esse comando é possível ver que foi instalado a versão PostgreSQL 13.

Após muitos testes, entre criar um usuário e um banco, consegui finalizar a instalação do OpenCMS e constatou-se que não foi necessária essa etapa, por esse motivo não vou colocá-la aqui. Para que funcione neste momento é fundamental somente colocar senha no usuário padrão do PostgreSQL, que é o usuário “postgres” gerado automaticamente e tem permissão total dentro do banco de dados, no Debian ele não vem com nenhuma senha padrão, por esse motivo foi necessário adicionar uma. Primeiro se promova ao usuário postgres e entre no terminal com os comandos:

|  |
| --- |
| su - postgres |

Seguido de:

|  |
| --- |
| psql |

Essa parte pode ser possível ser feita com outros comandos, porém estes foram os utilizados. Dentro do psql vai ser possível ver escrito “postgres=#” que expressa que você está dentro do terminal, em seguida digite esse comando:

|  |
| --- |
| ALTER USER postgres WITH PASSWORD ‘suaSenhaSegura123’; |

Troque a senha por uma de sua preferência. Após executar o comando, se for bem-sucedido, irá aparecer “ALTER ROLE”.



Depois digite um “\q” para sair do terminal. Ainda no usuário “postgres” digite:

|  |
| --- |
| psql -U postgres -h localhost -W |

Este comando tenta entrar no usuário postgres utilizando a senha, para testar se funcionou. Se entrar no usuário significa que funcionou, digite um “\q” e em seguida um “exit” para voltar ao usuário root.

* Agora é a instalação do Tomcat 9. Para instalar o servidor de aplicações utilize o comando:

|  |
| --- |
| apt install -y tomcat9 |

Em seguida, para instalar o painel administrativo que faz o gerenciamento de usuários e aplicações, entre outros, utilize o comando:

|  |
| --- |
| apt install -y tomcat19-admin |

Depois verifique o status do serviço com:

|  |
| --- |
| systemctl status tomcat9 |

Se todas as condições forem atendidas tente o acesso em seu navegador com: http://IP\_DA\_VM:8080.

* Depois de tudo configurado e em funcionamento é preciso fazer o *deploy* do OpenCMS dentro do Tomcat, e completar a instalação via navegador. Na documentação diz que o OpenCMS pode ser implantado copiando e colando o arquivo “.war” dentro do diretório “webapps/” do Tomcat. Se você copiar ele com o nome padrão “opencms.war” o OpenCMS poderá ser acessado com o “/opencms” ao final do IP, e para se livrar dessa questão basta renomear o arquivo “opencms.war” para “ROOT.war” antes de copiar o arquivo para o diretório do Tomcat. Desta forma o OpenCMS poderá ser acessado como aplicativo raiz. Além de ser possível alterar a porta no “/etc/tomcat9/server.xml”, o arquivo de configuração do Tomcat.

No meu caso eu fui para a pasta \home com o comando:

|  |
| --- |
| cd /home |

E baixei a versão estável do OpenCMS que no caso será a 13.0.0 com o comando:

|  |
| --- |
| wget https://github.com/alkacon/opencms-core/releases/download/build\_13\_0\_0/opencms-13.0-distribution.zip |

O comando “wget” funciona para baixar arquivos da internet. Após instale um aplicativo para descompactar esse zip, nesse caso será usado o aplicativo unzip, instale ele com o comando:

|  |
| --- |
| apt install unzip -y |

Depois extraia com o comando:

|  |
| --- |
| unzip opencms-19.0-distribution.zip |

Copie o .war para o arquivo webapps do Tomcat com o comando:

|  |
| --- |
| cp opencms.war /var/lib/tomcat9/webapps/ |

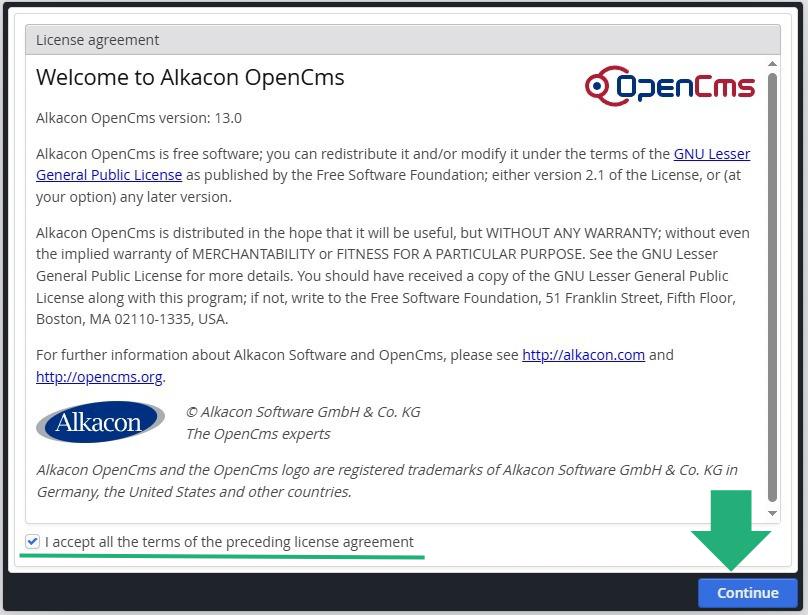
Reinicie o Tomcat com o comando:

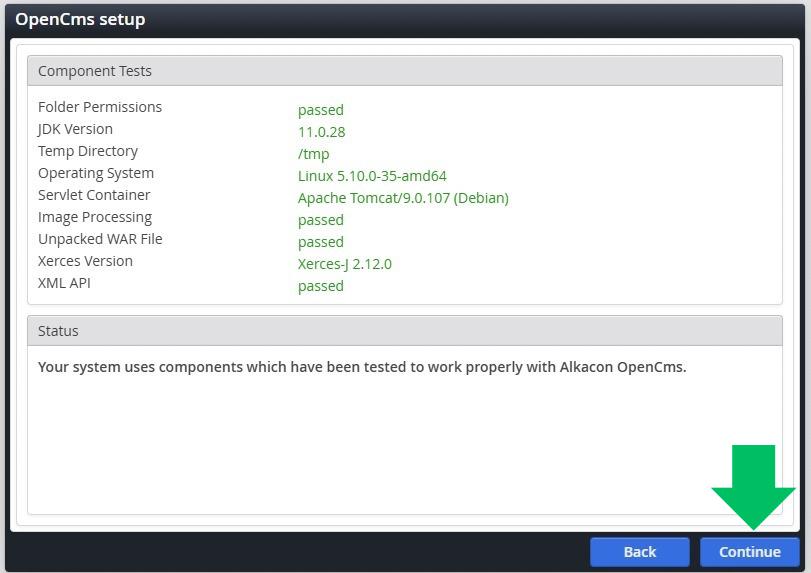
|  |
| --- |
| systemctl restart tomcat9 |

E abra o navegador para conferir se funcionou com o comando:

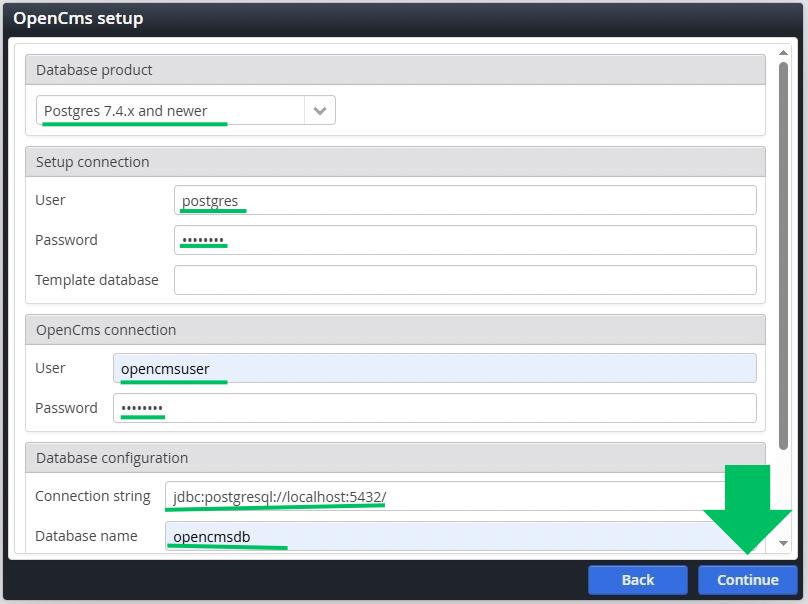
|  |
| --- |
| http://IPDAVM:8080/opencms/setup |

Fui para a pasta /home devido ela ter mais espaço para fazer o download do .zip OpenCMS. Primeiramente iria fazer na pasta /tmp para os arquivos serem removidos sozinhos, porém, em minha VM, não havia espaço nessa partição para fazer o download o que ocasionou alguns erros, pois o arquivo ficou corrempido, e para evitar isso que foi utilizado o /home, e logo depois os outros arquivos que vinham junto no download foram excluídos manualmento com o comando “rm”.

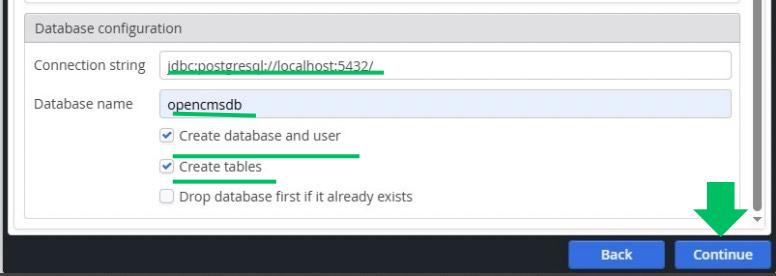
* Ao entrar na configuração do OpenCSM a primeira tela a ser vista é o contrato de licença, após marcar a opção que aceita os termos só é preciso clicar em “Continue”.



A próxima etapa é a de verificação do sistema, se tiver seguido todas as etapas até aqui é para todos os componentes estarem corretos e com a coloração esverdeada. Clique em “Continue” novamente e será levado para a configuração do banco de dados.



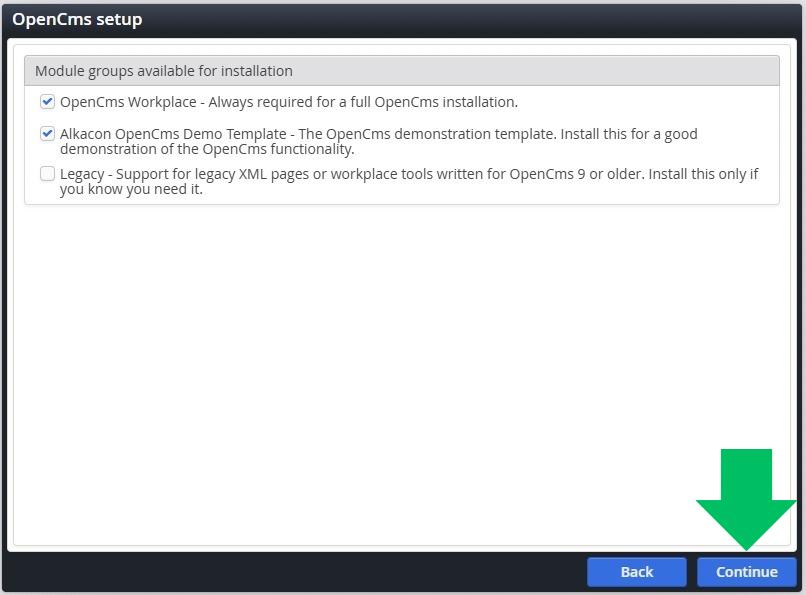
No local onde consta “Database product” selecione o banco, que no nosso caso é a opção: “Postgres 7.4.x and newer”. Em “Setup connection” se deve colocar o usuário postgres e a senha que foi criada anteriormente, o que proporciona a opção de template em branco. Em “OpenCMS connection” coloque o nome do usuário que será criado dentro do banco e sua senha.



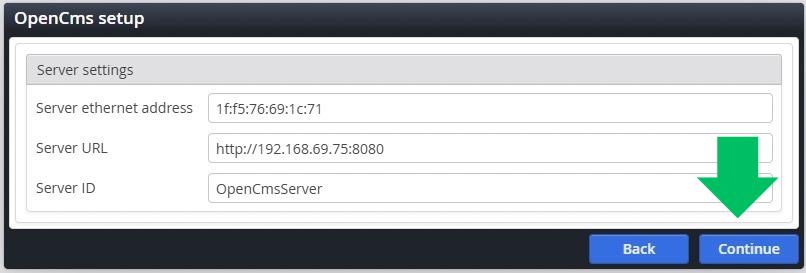
Agora em “Database configuration” no primeiro campo deixe como está, pois todas as informações são padrão dentro do PostgreSQL, desde o nome a porta usada, mas para ter certeza o conteúdo que deve constar no campo é: “jdbc:postgresql://localhost:5432/”. E no segundo campo coloque o nome do banco que será criado, deixe as duas opções marcadas, para ser criado o banco, o usuário e as tabelas. E depois clique novamente em “Continue”.

As informações de “Setup connection” são utilizadas apenas durante a configuração, especificamente na criação das tabelas e o esquema do banco de dados. Nesse momento é preciso permissões totais para o banco de dados, essas informações utilizadas não são persistidas em nenhum lugar, sendo descartadas após o final da configuração.

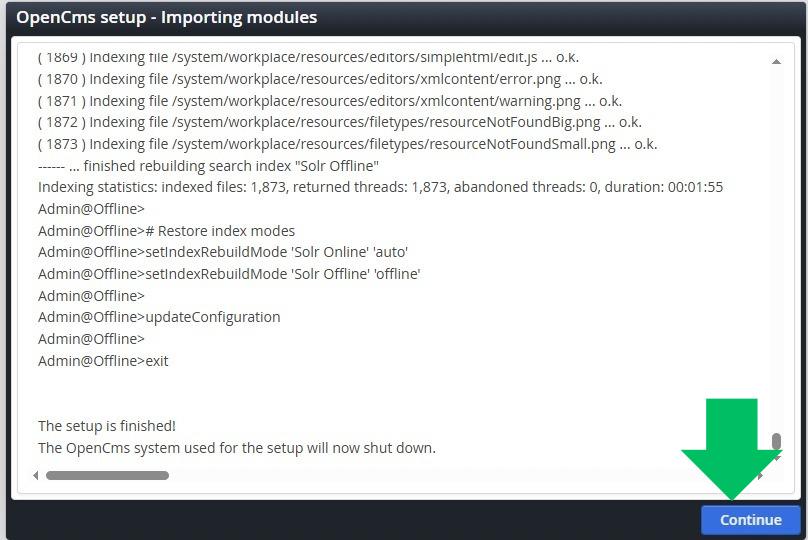
As informações de “OpenCMS connection” são utilizadas para todas operações com o banco durante a instalação, não requer permissões totais para o banco, somente permisões de leitura e gravação. Essas informações são colocadas no arquivo “opencms.properties” que fica no diretório do Tomcat: “{CATALINA\_HOME}/webapps/<OPENCMS>/config/”. Onde poderão ser alteradas posteriormente caso necessário.



Nesta próxima etapa você pode deixar as opções marcadas, a primeira opção é necessária para uma instalação completa do OpenCMS que somente não seria fundamental em uma instalação em cluster com replicação a onde somente o servidor precisa desses módulos. Já o segundo seria o *template* com o objetivo de evidenciar algumas funcionabilidades do OpenCMS.

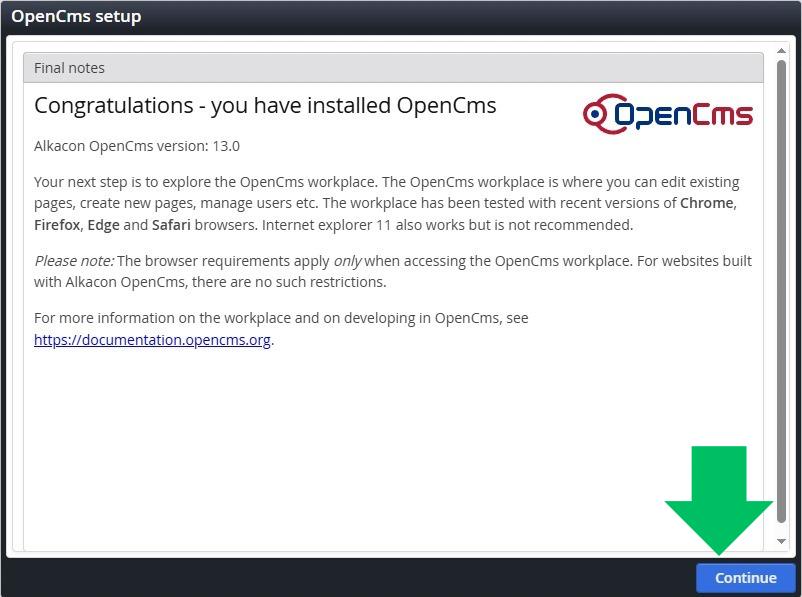


Este próximo passo que consiste em nada ser feito, pois é mais específica para uma instalação em cluster.

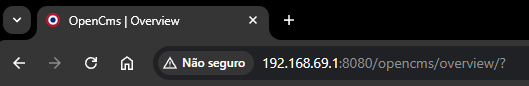


Na etapa subsequente seria somente a instalação dos módulos, que pode levar um tempo a depender da estrutura criada para a VM.

Em seguida é mostrada a tela final, e você, após clicar e “Continue”.



Após finalizar a instalação do OpenCMS ele te redireciona para a página mostrada a baixo:



* Agora, para concluir, realiza-se a instalação do Nginx para funcionar como Proxy Reverso dentro de nossa estrutura. Antes da instalação utilizei o comando:

|  |
| --- |
| apt update && apt upgrade |

No meu caso, havia um arquivo para ser atualizado, e depois seria a instalação do Nginx com o comando:

|  |
| --- |
| apt install nginx -y |

Com a instalação do Nginx ele vai receber as requisições HTTP externas na porta 80 e repassar para o Tomcat. Depois da instalação será indispensável criar um bloco de configuração para o site na pasta “sites-available” com o comando:

|  |
| --- |
| nano /etc/nginx/sites-available/opencms |

Esse comando inicia o arquivo de texto “opencms” que será preenchido com a seguinte configuração:

|  |
| --- |
| server {  listen 80;  server\_name 192.168.69.70;  location / {  proxy\_pass http://localhost:8080/;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real\_IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  }  } |

* No arquivo de configuração em questão, consta-se:
* O “listen 80;” serve para que o nginx escute toda solicitação enviada na porta 80, que é a padrão HTTP.
* O “server\_name 192.168.69.70;” define o nome que o servidor responde.
* O “location / {}” define o que fazer para qualquer URL com os comandos que ficam dentro dele.
* O “proxy\_pass http://localhost:8080/;” encaminha as requisições para o Tomcat que está rodando localmente na porta 8080.
* O “proxy\_set\_header” passa algumas informações importantes para o Tomcat, que nesse caso são:
* Host: ele manda esse valor pois muitas aplicações dependem dele para construir links corretos, e não gerar URLs com a porta errada.
* X-Real\_IP: envia o IP real do cliente para o Tomcat/OpenCMS.
* X-Forwarded\_For: basicamente, ele adiciona o IP do cliente no header “X-Forwarded\_For” sendo o padrão que muitos aplicativos e bibliotecas leem para saber o IP do cliente. Com o   
  “$proxy\_add\_x\_forwarded\_for” concatenando o IP atual.
* X-Forwarded-Proto: informa para o backend o esquema original da requisição, http ou https, pois futuramente pode ser útil para gerar redirecionamentos corretos e outras decisões de segurança.
* Haviam outras configurações que poderiam ser adicionadas, com outras funcionalidades específicas, mas que optei de não utilizar.

Após escrever o arquivo de configuração salve com “ctrl-o” e saia do arquivo com o “ctrl-x”. Logo depois será necessário ativar a configuração com o comando:

|  |
| --- |
| ln -s /etc/nginx/sites-available/opencms /etc/nginx/sites-enabled |

O nginx no Debian usa “sites-available” para guardar as configurações e “sites-enabled” para as que estão ativas, esse comando utiliza um link simbólico para ativar o site. Sendo “ln” para criar o link e “-s” para indicar que o link é simbólico. Caso haja necessidade de edição basta alterar o arquivo que foi criado e testar a sintaxe e reiniciar o serviço que as alterações já estarão presentes no link criado.

Também é recomendado que o arquivo padrão seja removido, que no meu caso, ocasionou um erro na hora de reiniciar o nginx, para isso utilize o comando:

|  |
| --- |
| rm /etc/nginx/sites-enabled/default |

Esse comando remove o arquivo padrão do site. Em seguida faça um teste da sintaxe de sua configuração com o comando:

|  |
| --- |
| nginx -t |

Se estiver tudo certo terá o retorno:



Depois reinicie o nginx:

|  |
| --- |
| systemctl restart nginx |

Agora é testar se está tudo em funcionamento no seu navegador utilizando: “http://IP\_DA\_VM/”, que assim será possível ver a página inicial “It Works!” do Tomcat, adicionando o “/opencms” ele já te encaminha para a página do Opencms. Dessa forma, concluo o laboratório.

Agradeço pela oportunidade concedida. Apesar dos diversos desafios enfrentados, foi possível concluir o trabalho, revisando conceitos previamente estudados e assimilando novos conhecimentos. Que seja apenas o início de um trajetória de aprendizado contínuo.

**Referências Bibliográficas**

**OPEN CMS. Getting started.** Disponível em: <https://documentation.opencms.org/opencms-documentation/introduction/get-started/>. Acesso em: 9 ago. 2025.

**APACHE SOFTWARE FOUNDATION. Host Manager How-To**. Disponível em: <https://tomcat.apache.org/tomcat-9.0-doc/host-manager-howto.html>. Acesso em: 9 ago. 2025.

**NGINX. Documentation**. Disponível em: <https://nginx.org/en/docs/>. Acesso em: 10 ago. 2025.