

INSTITUTO INFNET

Projeto de Bloco: Arquitetura de Infraestrutura de Aplicações

Flavio Thomaz Junior



TESTE DE PERFORMANCE 1

Sumário

TESTE DE PERFORMANCE 1 3

TESTE DE PERFORMANCE 2 6

TESTE DE PERFORMANCE 3 8

TESTE DE PERFORMANCE 4 11

TESTE DE PERFORMANCE 5 14

TESTE DE PERFORMANCE 6 17

TESTE DE PERFORMANCE 7 21

TESTE DE PERFORMANCE 8 27

CONCLUSÕES..... 30

TESTE DE PERFORMANCE 1

Introdução

A empresa xabaraz investimentos atua no mercado há mais de 40 anos realizando consultoria de aplicações financeiras na bolsa de valores, além de fornecer suporte a seus clientes na escolha das melhores ações, rendimentos futuros e o alertando para os eventuais riscos que possam existir. Com solidez conquistada perante o mercado, a empresa se encontra atualmente com um quadro de 1200 funcionários, atuando em diferentes frentes, para fornecer a seus clientes o melhor atendimento no menor tempo possível.

Dado a expansão dos negócios ao longo dos anos, foram necessárias consecutivas intervenções na sua infraestrutura, para modernizar a empresa com as tecnologias vigentes e aumentar seu potencial de produção. Apesar disso, a empresa se encontra com dificuldades para encontrar uma solução que permita integrar suas equipes de uma maneira completa, oferecendo opções de compartilhamento de arquivos, mensagens instantâneas, comunicação por áudio/vídeo dentre outras. Os gestores estão cansados de ter que usar múltiplas aplicações para diferentes propósitos, e solicitaram ao setor de TI a implementação de uma única solução para lidar com tudo.

Este problema tem impactado a produtividade dos funcionários pois os mesmos por muitas vezes tem de lidar com múltiplos logins e senhas para diferentes aplicações, tem de deixar várias abertas ao mesmo tempo em suas máquinas (o que dificulta agilidade de operação entre elas), além de ser demasiado confuso ter uma aplicação para cada tarefa desejada. Uma solução all-in-one permitiria a empresa fornecer a seus colaboradores uma plataforma simplificada com que possam se comunicar e se integrar aos negócios da empresa, em tempo real e sem a necessidade de ter de buscar soluções de terceiros para isso.

A respeito da Aplicação

O nextcloud é uma solução de código aberto, totalmente gratuita, que tem por finalidade construir um ambiente de nuvem utilizando a infraestrutura on premise já existente de uma organização. É uma aplicação com solidez no mercado, já possui modelos de implementação tanto em nuvem pública, quanto em VM e em containers, cada uma com suas vantagens e desvantagens. No presente documento veremos algumas características de sua infraestrutura, o modelo que deve ser implementado na organização e de que forma pode-se manipular o seu código visando extrair uma experiência de qualidade personalizada.

O nextcloud possui dois componentes chaves para seu funcionamento:

- um banco de dados (nesse caso, MariaDB)
- um servidor nginx (servidor web)

A aplicação já possui modelos de instalação pronto para uso, bastando que se configure alguns detalhes como usuário e senha após a conclusão da mesma. Por se tratar de uma aplicação distribuída, já que possui dependências já mencionadas para funcionar propriamente, pode ser considerada uma SDDC.

A Respeito da Implementação

Partindo do princípio que o objetivo principal de implementar a mencionada aplicação é reduzir o esforço administrativo do cotidiano do trabalho, deve-se pensar em uma forma de aplica-la na empresa sem que demande de tarefas demasiado complexas. Portanto, a implementação da aplicação se dará da seguinte forma:

- I. Será usada a infraestrutura já existente, e usada para instalar uma imagem do Ubuntu Server, que será o sistema onde a aplicação ficará hospedada.
- II. A aplicação rodará em cima de containers, utilizando a solução de Docker para isso
- III. Será usado um modelo de instalação automática, através de um playbook ansible, que instalará todas as dependências e os devidos containers necessários

A aplicação rodando em cima de containers estaremos reduzindo o esforço administrativo tanto para sua instalação quanto para sua eventual manutenção, visto que ela rodaria através de três containers que atuariam em conjunto:

- O banco de dados (MariaDB)
- O servidor web (Nginx)
- A aplicação em si

Caso houvesse algum problema, ele poderia ser investigado individualmente, ou seja, não seria necessário uma intervenção na aplicação como um todo, mas apenas em suas partes.

TESTE DE PERFORMANCE 2

Instalação do nextcloud pelo ansible

O nextcloud consiste em um ambiente de nuvem on-premise, onde pode-se hospedar toda uma infraestrutura local de armazenamento, e compartilha-la com os clientes. Por conta disso, existem vários fatores envolvidos em sua instalação, como o servidor web, o banco de dados, a configuração php. E através do ansible, todas essas dependências podem ser facilmente instaladas com o mínimo de configuração possível. Portanto, seguem-se os passos executados para a instalação dessa aplicação:

1. Preparar o ambiente Linux para a instalação da aplicação e a instalação do ansible:
curl -s https://raw.githubusercontent.com/ReinerNippes/nextcloud/master/prepare_system.sh | /bin/bash

O comando curl serve para verificar a conectividade da url e o parâmetro -s especifica que deve ser realizada uma transferência de dados entre o link fornecido e a máquina.

```
ubuntu@ubuntu: ~/nextcloud
File Edit View Search Terminal Help
ubuntu@ubuntu:~$ curl -s https://raw.githubusercontent.com/ReinerNippes/nextcloud/master/prepare_system.sh | /bin/bash
Hit:1 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://ppa.launchpad.net/remmina-ppa-team/remmina-next/ubuntu bionic InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
281 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.3.0-51 linux-headers-5.3.0-51-generic
  linux-image-5.3.0-51-generic linux-modules-5.3.0-51-generic
  linux-modules-extra-5.3.0-51-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  i965-va-driver libavcodec57 libavutil55 libcacard0 libcrystalhd3 libgsm1
  libnetplan0 libopenjp2-7 libphodav-2.0-0 libphodav-2.0-common libshine3
  libsnappy1v5 libsoxr0 libspice-client-glib-2.0-8 libspice-client-gtk-3.0-5
  libswresample2 libusbredirhost1 libusbredirparser1 libva-drm2 libva-x11-2
  libva2 libvdpau1 libx264-152 libx265-146 libxvidcore4 libzvb1-common
  libzvb10 mesa-va-drivers mesa-va-driver-all python3-dateutil
  remmina-plugin-spice spice-client-glib-usb-acl-helper va-driver-all
  vdpau-driver-all xdg-desktop-portal xdg-desktop-portal-gtk
The following packages will be upgraded:
```

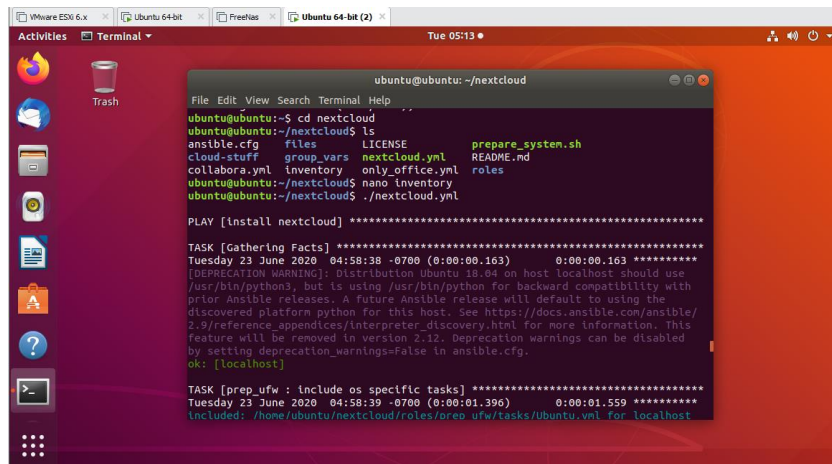
2. Clonar o repositório do nextcloud: git clone
<https://github.com/ReinerNippes/nextcloud>

```
ubuntu@ubuntu:~$ git clone https://github.com/ReinerNippes/nextcloud
Cloning into 'nextcloud'...
remote: Enumerating objects: 2998, done.
remote: Total 2998 (delta 0), reused 0 (delta 0), pack-reused 2998
Receiving objects: 100% (2998/2998), 845.67 KiB | 11.75 MiB/s, done.
Resolving deltas: 100% (1642/1642), done.
```

Esse comando irá realizar uma copia de todo o repositório do nextcloud para dentro da máquina

3. Para prosseguir iremos editar o arquivo inventory na pasta nextcloud, com o comando nano, para definir as credenciais de acesso a página web.

4. Com o arquivo devidamente editado, prosseguiremos com a instalação através do comando “./nextcloud.yml”. Esse comando irá executar nosso playbook e instalar/configurar todas as roles necessárias para a instalação do nextcloud
- Roles usadas na instalação

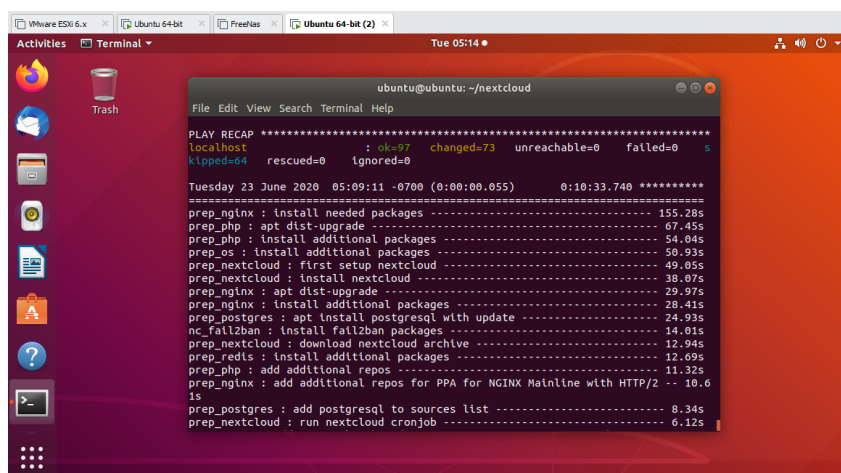


```
ubuntu@ubuntu: ~/nextcloud
File Edit View Search Terminal Help
ubuntu@ubuntu:~$ cd nextcloud
ubuntu@ubuntu:~/nextcloud$ ls
ansible.cfg  files      LICENSE    prepare_system.sh
cloud-stuff  group_vars nextcloud.yml  README.md
collabora.yml  inventory only_office.yml  roles
ubuntu@ubuntu:~/nextcloud$ nano inventory
ubuntu@ubuntu:~/nextcloud$ ./nextcloud.yml

PLAY [install nextcloud] *****

TASK [Gathering Facts] *****
Tuesday 23 June 2020  04:58:38 -0700 (0:00:00.163)    0:00:00.163 *****
[Deprecation Warning]: Distribution Ubuntu is not on host localhost should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A future Ansible release will default to using the
discovered platform python for this host. See https://docs.ansible.com/ansible/
2.9/reference_appendices/interpreter_discovery.html for more information. This
feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
ok: [localhost]

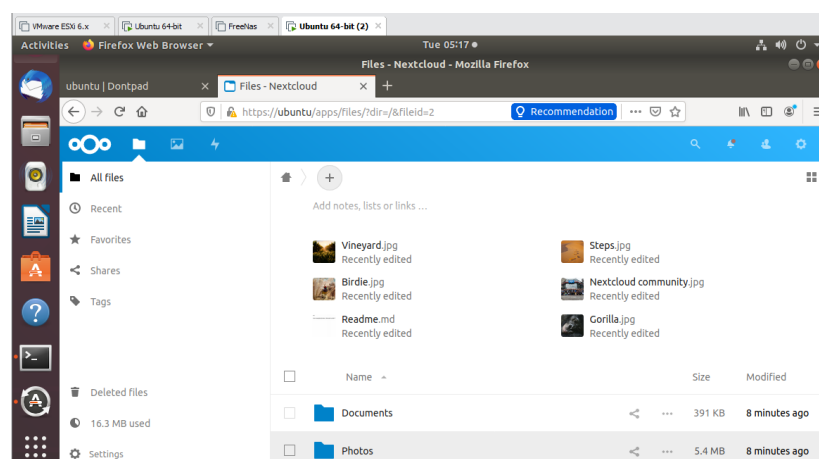
TASK [prep_ufw : include os specific tasks] *****
Tuesday 23 June 2020  04:58:39 -0700 (0:00:01.396)    0:00:01.559 *****
included: /home/ubuntu/nextcloud/roles/prep_ufw/tasks/Ubuntu.yml for localhost
```



```
PLAY RECAP *****
localhost : ok=97  changed=73  unreachable=0  failed=0  s
kipped=04  rescued=0  ignored=0

Tuesday 23 June 2020  05:09:11 -0700 (0:00:00.055)    0:10:33.740 *****
=====
prep_nginx : install needed packages ..... 155.28s
prep_php : apt dist-upgrade ..... 67.45s
prep_php : install additional packages ..... 54.04s
prep_php : install additional packages ..... 50.93s
prep_nextcloud : first setup nextcloud ..... 49.05s
prep_nextcloud : install nextcloud ..... 38.07s
prep_nginx : apt dist-upgrade ..... 29.97s
prep_nginx : install additional packages ..... 28.41s
prep_postgres : apt install postgresql with update ..... 24.93s
nc_fail2ban : install fail2ban packages ..... 14.01s
prep_nextcloud : download nextcloud archive ..... 12.94s
prep_redis : install additional packages ..... 12.69s
prep_php : add additional repos ..... 11.32s
prep_nginx : add additional repos for PPA for NGINX Mainline with HTTP/2 .. 10.6
1s
prep_postgres : add postgresql to sources list ..... 8.34s
prep_nextcloud : run nextcloud cronjob ..... 6.12s
```

5. Após a execução do playbook, basta acessar a url determinada no script para acessar a aplicação:



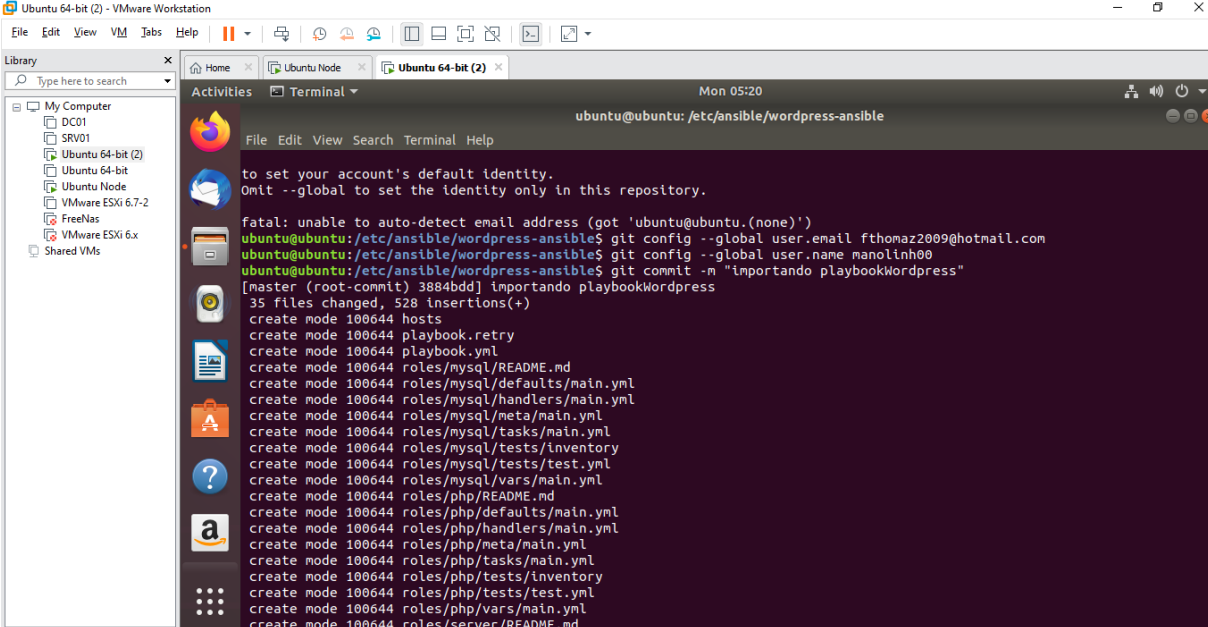
TESTE DE PERFORMANCE 3

Criação de Repositório

Na instalação de aplicativos, criar um repositório significa armazenar todas as informações relacionadas a ele, desde os arquivos de configuração até a explicação de como deve ser executado a instalação (a exemplo do arquivo README). Um repositório pode ser utilizado na instalação de uma aplicação em múltiplas maquinas, pois os arquivos encontram-se online e disponíveis para realizar uma automatização através de ferramentas como o ansible ou semelhantes. Para fins de explicação, o presente documento apresentará os prints do processo de criação de um repositório playbook ao github, bem como a explicação devida do que ocorre em cada tela.

Criação do repositório

Para a criação do repositório, primeiramente deve-se configurar as informações pertinentes a ele, como o login e senha a serem utilizados para acessá-lo, para que seja possível posteriormente fazer o upload do playbook para ele. Abaixo o print da configuração realizada:



The screenshot shows a terminal window titled 'Ubuntu 64-bit (2) - VMware Workstation'. The terminal is running commands to configure a git repository. The output shows the creation of a new repository, configuration of user email and name, and a commit of the 'playbookWordpress' directory. The terminal output is as follows:

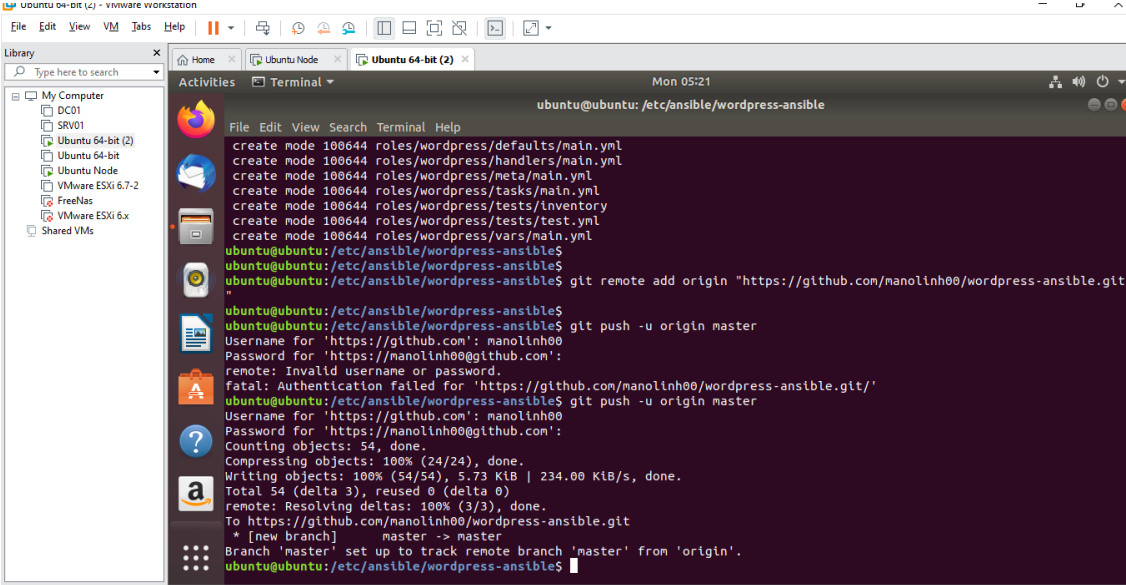
```
ubuntu@ubuntu: /etc/ansible/wordpress-ansible
to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'ubuntu@ubuntu.(none)')
ubuntu@ubuntu:/etc/ansible/wordpress-ansible$ git config --global user.email fthomaz2009@hotmail.com
ubuntu@ubuntu:/etc/ansible/wordpress-ansible$ git config --global user.name manolinh00
ubuntu@ubuntu:/etc/ansible/wordpress-ansible$ git commit -m "importando playbookWordpress"
[master (root-commit) 3884bdd] importando playbookWordpress
35 files changed, 528 insertions(+)
create mode 100644 hosts
create mode 100644 playbook.retry
create mode 100644 playbook.yml
create mode 100644 roles/mysql/README.md
create mode 100644 roles/mysql/defaults/main.yml
create mode 100644 roles/mysql/handlers/main.yml
create mode 100644 roles/mysql/meta/main.yml
create mode 100644 roles/mysql/tasks/main.yml
create mode 100644 roles/mysql/tests/inventory
create mode 100644 roles/mysql/tests/test.yml
create mode 100644 roles/mysql/vars/main.yml
create mode 100644 roles/php/README.md
create mode 100644 roles/php/defaults/main.yml
create mode 100644 roles/php/handlers/main.yml
create mode 100644 roles/php/meta/main.yml
create mode 100644 roles/php/tasks/main.yml
create mode 100644 roles/php/tests/inventory
create mode 100644 roles/php/tests/test.yml
create mode 100644 roles/php/vars/main.yml
create mode 100644 roles/server/README.md
```

Como pode ser observado na imagem, o comando “git config” está realizando a função de configuração das credenciais do repositório, enquanto o “git commit” está importando o playbook do nextcloud para o mesmo.

Fazendo o upload do playbook para o repositório

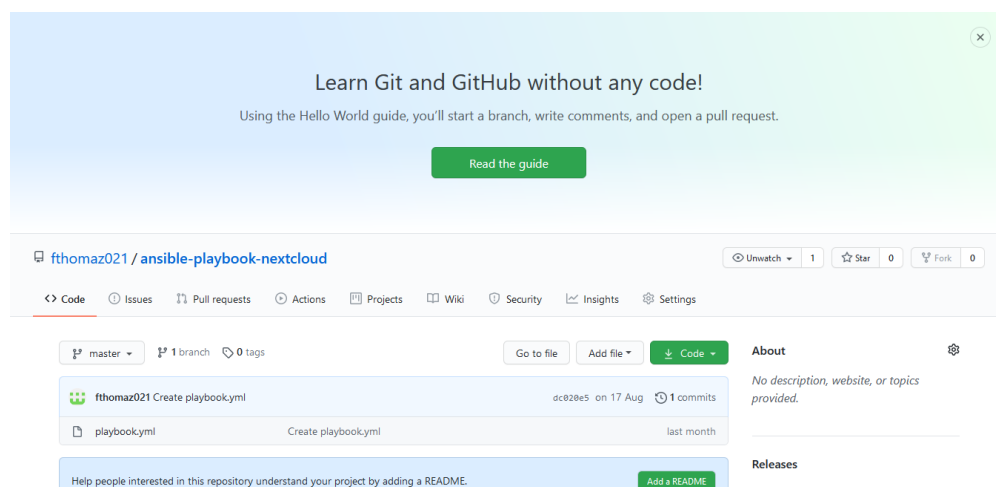
Após ter criado o repositório no github, e configurado através do parâmetro “git config” no terminal do Linux, o playbook foi enviado ao mesmo, conforme mostra o print abaixo:



```
ubuntu@ubuntu: /etc/ansible/wordpress-ansible
create mode 100644 roles/wordpress/defaults/main.yml
create mode 100644 roles/wordpress/handlers/main.yml
create mode 100644 roles/wordpress/meta/main.yml
create mode 100644 roles/wordpress/tasks/main.yml
create mode 100644 roles/wordpress/tests/inventory
create mode 100644 roles/wordpress/tests/test.yml
create mode 100644 roles/wordpress/vars/main.yml
ubuntu@ubuntu: /etc/ansible/wordpress-ansible$
ubuntu@ubuntu: /etc/ansible/wordpress-ansible$ git remote add origin "https://github.com/manolinh00/wordpress-ansible.git"
ubuntu@ubuntu: /etc/ansible/wordpress-ansible$
ubuntu@ubuntu: /etc/ansible/wordpress-ansible$ git push -u origin master
Username for 'https://github.com': manolinh00
Password for 'https://manolinh00@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/manolinh00/wordpress-ansible.git/'
ubuntu@ubuntu: /etc/ansible/wordpress-ansible$ git push -u origin master
Username for 'https://github.com': manolinh00
Password for 'https://manolinh00@github.com':
Counting objects: 54, done.
Compressing objects: 100% (24/24), done.
Writing objects: 100% (54/54), 5.73 KiB | 234.00 KiB/s, done.
Total 54 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/manolinh00/wordpress-ansible.git
 * [new branch] master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
ubuntu@ubuntu: /etc/ansible/wordpress-ansible$
```

Conforme exposto, foi utilizado o comando “git remote add origin” para indicar o link aonde o playbook seria enviado. Após isso, o comando “ git -u origin master” informa que o playbook deve ser enviado ao repositório anteriormente configurado, sendo enviado ao link informado.

Com todos os passos realizados, o playbook já se encontra dentro do repositório do github:



Ele se encontra na seguinte pagina: <https://github.com/fthomaz021/ansible-playbook-nextcloud>

TESTE DE PERFORMANCE 4

Aplicação a ser executada

O nextcloud trata-se de um conjunto de software cliente-servidor para criar e usar serviços de hospedagem de arquivos, de código aberto e 100% gratuito. É uma solução de nuvem privada open source que permite o armazenamento e sincronização de arquivos em múltiplos dispositivos, podendo ser instalado tanto em ambiente físico (em um servidor) quanto em kubernetes (através do docker). A vantagem por trás do nextcloud é a possibilidade de se utilizar uma infraestrutura pré-existente de uma empresa para montar uma nuvem privada sem incorrer em custos de licenciamento de software. O nextcloud possui ferramentas para comunicação de equipes (com possibilidade de chamadas de áudio/vídeo), editor de documentos, ferramenta de grupo que possibilita integrar calendários, e-mails e contatos para ajudar as equipes a terem um ambiente de trabalho mais produtivo.

Por se tratar de uma solução que possibilita a criação de nuvem privada, o nextcloud se utiliza de banco de dados, servidor apache, php. Portanto, qualificando-se como uma aplicação distribuída, pois ela pode ser instalada e acessada em múltiplos dispositivos, desde computadores até dispositivos moveis como tablets e smartphones. A utilização da mesma como solução de nuvem privada incorre nos seguintes benefícios:

1. Ampla documentação disponível para implantação em ambientes on premises, tanto no site oficial da solução, quanto nas comunidades web
2. Possibilidade de utilizar contêiner para sua instalação, tendo uma imagem oficial no Docker
3. Solução 100% open source, o que permite que seja auditada independentemente a procura de quaisquer brechas de segurança além de modificada conforme as necessidades do cliente/empresa.
4. Tem como missão oferecer uma plataforma para o compartilhamento de arquivos e integração de equipes que esteja no controle do cliente, tendo ele total responsabilidade sobre o que ocorre em seu ambiente, não tendo que confiar em terceiros para isso:

“O nextcloud coloca o cliente no controle de seus dados no sentido mais literal e direto. Seus dados estão em seu centro de dados, em um servidor gerenciado por você, em vez de flutuarem em algum lugar na nuvem.”

Processo de instalação

Para realizar a instalação do nextcloud, foi utilizado:

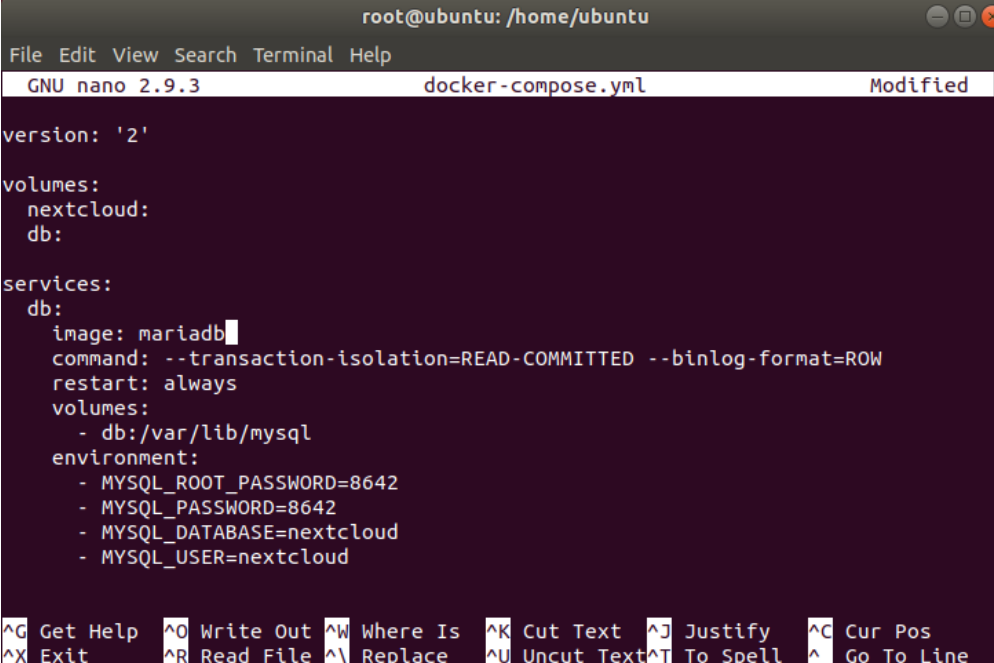
- Uma máquina virtual Linux Ubuntu
- Docker-compose e Docker.io
- Um container contendo banco de dados (MariaDB)
- Um container contendo servidor web(Nginx)
- Um container contendo a aplicação (Nextcloud)

Foram usadas três linhas de comando: duas para a instalação do docker propriamente dito, e a última para o download e configuração da ferramenta nextcloud através do Docker-compose. Foi criado um script (docker-compose.yml) instalando todas as dependências da aplicação (como banco de dados e servidor web).

1- apt install docker.io

2- apt install docker.io docker docker-compose

3- Editar o documento docker-compose.yml com as instruções para a instalação do nextcloud, do banco de dados(MariaDB) e o servidor WEB Nginx, conforme o print abaixo



```
root@ubuntu: /home/ubuntu
File Edit View Search Terminal Help
GNU nano 2.9.3 docker-compose.yml Modified

version: '2'

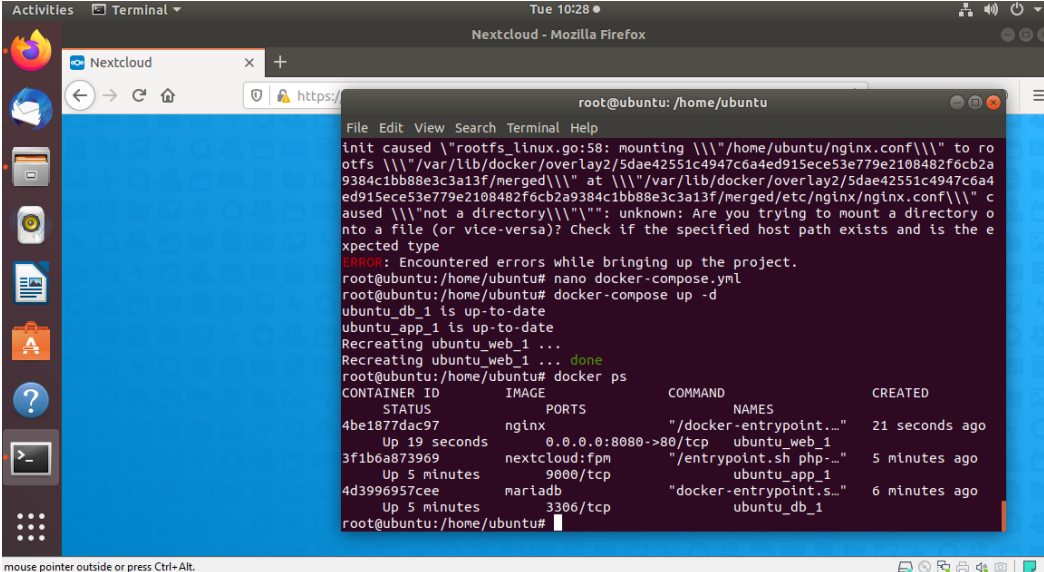
volumes:
  nextcloud:
  db:

services:
  db:
    image: mariadb
    command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
    restart: always
    volumes:
      - db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=8642
      - MYSQL_PASSWORD=8642
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

4- Executar o comando docker-compose up -d para executar o arquivo anteriormente instalado

5- Após a instalação, com o comando docker ps será possível identificar que estão rodando três containers: o banco de dados MariaDB, o servidor web Nginx e o nextcloud propriamente dito.



```
root@ubuntu: /home/ubuntu
File Edit View Search Terminal Help
init caused \"/rootfs_linux.go:58: mounting \"/home/ubuntu/nginx.conf\" to ro
otfs \"/var/lib/docker/overlay2/5dae42551c4947c6a4ed915ece53e779e2108482f6cb2a
9384c1bb88e3c3a13f/merged\" at \"/var/lib/docker/overlay2/5dae42551c4947c6a4
ed915ece53e779e2108482f6cb2a9384c1bb88e3c3a13f/merged/etc/nginx/nginx.conf\" c
aused \"/not a directory\" \"/\": unknown: Are you trying to mount a directory o
nto a file (or vice-versa)? Check if the specified host path exists and is the e
xpected type
ERROR: Encountered errors while bringing up the project.
root@ubuntu:/home/ubuntu# nano docker-compose.yml
root@ubuntu:/home/ubuntu# docker-compose up -d
ubuntu_db_1 is up-to-date
ubuntu_app_1 is up-to-date
Recreating ubuntu_web_1 ...
Recreating ubuntu_web_1 ... done
root@ubuntu:/home/ubuntu# docker ps
CONTAINER ID        IMAGE               COMMAND                  NAMES                  CREATED
4be1877dac97       nginx              "/docker-entrypoint..." ubuntu_web_1           21 seconds ago
3f1b6a873969       nextcloud:fpn      "/entrypoint.sh php-..." ubuntu_app_1           5 minutes ago
4d3996957cee       mariadb            "docker-entrypoint.s..." ubuntu_db_1            6 minutes ago
root@ubuntu:/home/ubuntu#
```

6- Como a imagem acima mostra, basta que se acesse o localhost no navegador web para se ter acesso ao nextcloud

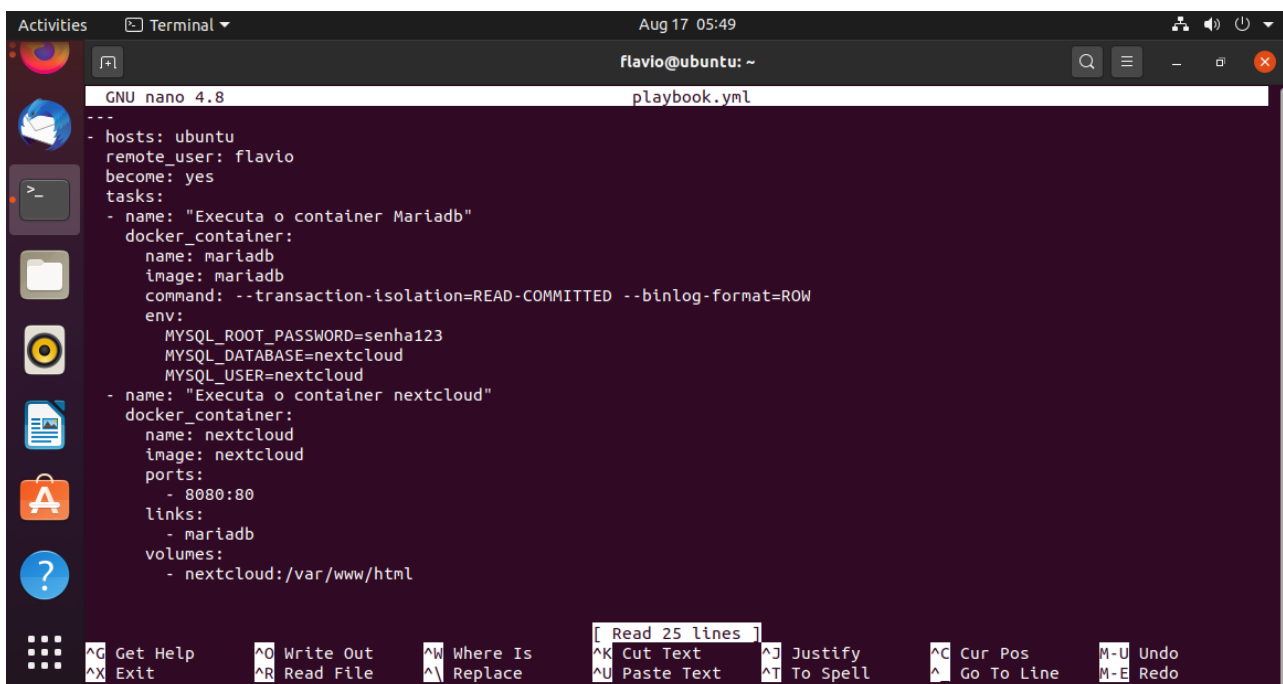
TESTE DE PERFORMANCE 5

Aplicação Distribuída através do ansible

O nextcloud é uma aplicação de armazenamento de arquivos na nuvem, de código aberto, e que permite sua instalação de diferentes formas, sendo uma delas através de containers. Para simplificar a sua utilização, bem como diminuir o esforço administrativo para sua configuração, serão utilizados dois containers:

- Banco de dados (MariaDB)
- Nextcloud com apache incluso

A vantagem aqui se dá primeiramente no fato de que através desse mecanismo a aplicação já vem pré-configurada, bastando realizar ajustes quanto as configurações de login, especificações do banco de dados (possível utilização de storage, replicação etc). Para automatizar esse processo, será utilizado um playbook ansible, que permitirá a instalação da aplicação sem a necessidade de muita intervenção manual para isso. Portanto, segue-se abaixo o script utilizado para tal:



```
GNU nano 4.8                                playbook.yml
---
- hosts: ubuntu
  remote_user: flavio
  become: yes
  tasks:
    - name: "Executa o container Mariadb"
      docker_container:
        name: mariadb
        image: mariadb
        command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
        env:
          MYSQL_ROOT_PASSWORD=senha123
          MYSQL_DATABASE=nextcloud
          MYSQL_USER=nextcloud
    - name: "Executa o container nextcloud"
      docker_container:
        name: nextcloud
        image: nextcloud
        ports:
          - 8080:80
        links:
          - mariadb
        volumes:
          - nextcloud:/var/www/html

[ Read 25 lines ]
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line    M-E Redo
```

Como mostra a figura acima, serão instalados os dois containers mencionados, com as senhas do banco de dados já definidas, além da porta em que a aplicação irá rodar, 8080.

Abaixo está o print do playbook rodando com sucesso:

The terminal window shows the execution of an Ansible playbook. The first part displays the output of the playbook, which includes gathering facts and executing tasks to start Docker containers. The second part shows the output of the 'docker ps' command, listing the running containers.

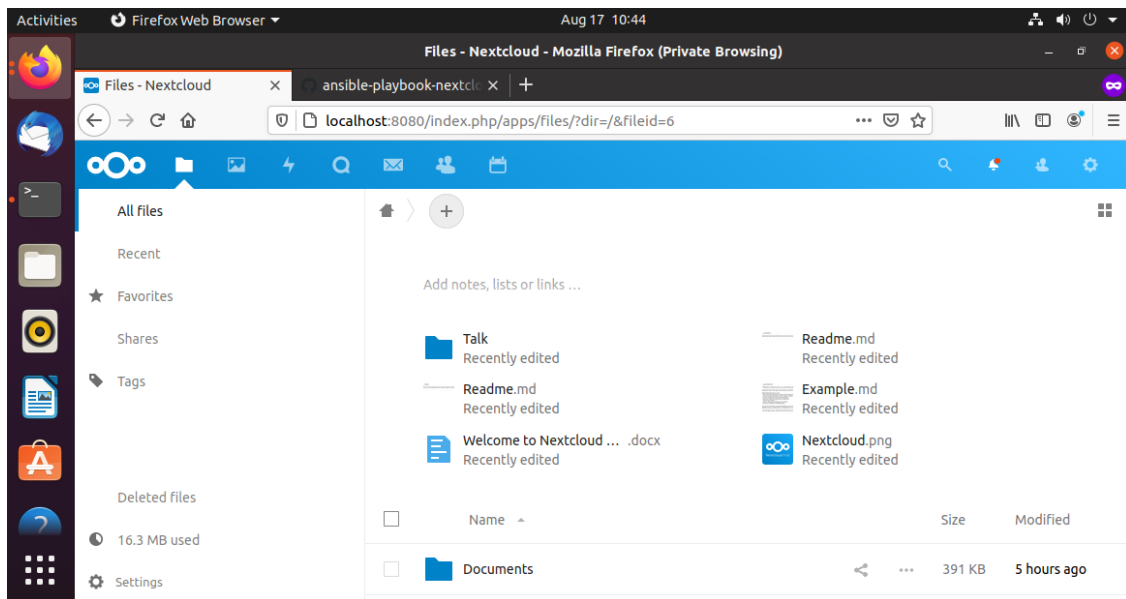
```
flavio@ubuntu: ~  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS  
72b3d8d6603e   mariadb   "docker-entrypoint.s..." About an hour ago Up About an hour 3306/tcp  
flavio@ubuntu:~$ ansible-playbook playbook.yml -i hosts.yml -K  
BECOME password:  
PLAY [ubuntu] *****  
TASK [Gathering Facts] *****  
ok: [192.168.159.132]  
TASK [Executa o container Mariadb] *****  
ok: [192.168.159.132]  
TASK [Executa o container nextcloud] *****  
changed: [192.168.159.132]  
PLAY RECAP *****  
192.168.159.132 : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0  
flavio@ubuntu:~$ sudo docker ps  
[sudo] password for flavio:  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS  
b0e1f1827117   nextcloud "/entrypoint.sh apac..." 22 minutes ago Up 22 minutes 0.0.0.0:8080->  
80/tcp  
72b3d8d6603e   mariadb   "docker-entrypoint.s..." 2 hours ago   Up 2 hours    3306/tcp  
flavio@ubuntu:~$
```

Para fins de melhor compreensão dos comandos, segue-se a explicação dos parâmetros utilizados:

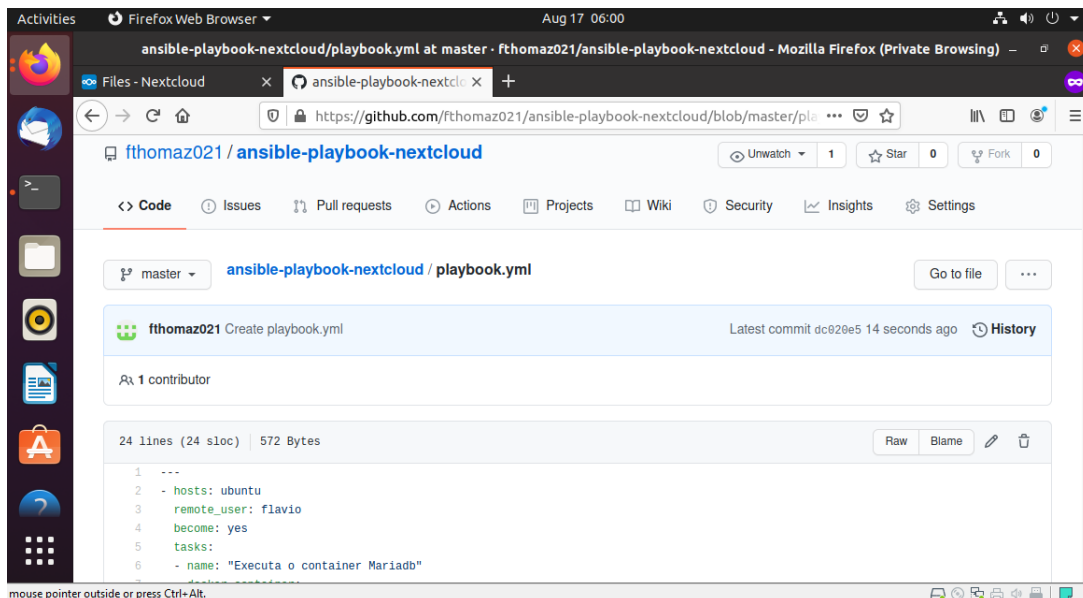
- `ansible-playbook`: comando para rodar o playbook com as instruções
- `playbook.yml`: arquivo contendo as instruções de instalação dos containers
- `-i`: Para fazer referência ao arquivo de hosts
- `hosts.yml`: arquivo com o host onde a aplicação será instalada
- `-K`: especifica que o comando será rodado com permissão de administrador (sudo)

Aplicação em Funcionamento

Com o playbook ansible executado, a aplicação já se mostra acessível através do navegador web, conforme o print abaixo:



Como forma de catalogar para posterior utilização, foi criado um repositório no github para alojar o playbook ansible usado na instalação, conforme mostra o print abaixo:



O link para o mesmo encontra-se no endereço: <https://github.com/fthomaz021/ansible-playbook-nextcloud/blob/master/playbook.yml>

TESTE DE PERFORMANCE 6

Introdução

A empresa xabaraz investimentos atua no mercado há mais de 40 anos realizando consultoria de aplicações financeiras na bolsa de valores, além de fornecer suporte a seus clientes na escolha das melhores ações, rendimentos futuros e o alertando para os eventuais riscos que possam existir. Com solidez conquistada perante o mercado, a empresa se encontra atualmente com um quadro de 1200 funcionários, atuando em diferentes frentes, para fornecer a seus clientes o melhor atendimento no menor tempo possível.

Dado a expansão dos negócios ao longo dos anos, foram necessárias consecutivas intervenções na sua infraestrutura, para modernizar a empresa com as tecnologias vigentes e aumentar seu potencial de produção. Apesar disso, a empresa se encontra com dificuldades para encontrar uma solução que permita integrar suas equipes de uma maneira completa, oferecendo opções de compartilhamento de arquivos, mensagens instantâneas, comunicação por áudio/vídeo dentre outras. Os gestores estão cansados de ter que usar múltiplas aplicações para diferentes propósitos, e solicitaram ao setor de TI a implementação de uma única solução para lidar com tudo.

Este problema tem impactado a produtividade dos funcionários pois os mesmos por muitas vezes tem de lidar com múltiplos logins e senhas para diferentes aplicações, tem de deixar várias abertas ao mesmo tempo em suas máquinas (o que dificulta agilidade de operação entre elas), além de ser demasiado confuso ter uma aplicação para cada tarefa desejada. Uma solução all-in-one permitiria a empresa fornecer a seus colaboradores uma plataforma simplificada com que possam se comunicar e se integrar aos negócios da empresa, em tempo real e sem a necessidade de ter de buscar soluções de terceiros para isso.

Sobre a Aplicação

O nextcloud trata-se de um ambiente integrado on-premise de colaboração. Ele tem como principal objetivo fornecer uma plataforma 100% de código aberto e que possibilite substituir diversas aplicações com uma única solução all-in-one, e gerenciado pela própria empresa, para que ela esteja no controle dos seus dados sem ter de delega-los a terceiros. Com ele, é possível:

- Sincronizar arquivos entre desktop, dispositivos móveis e interface web
- Se comunicar através de áudio/vídeo/texto através do nextcloud talk
- Integrar e-mails, grupos, contatos e calendários através do nextcloud groupware, que fornece ferramentas para times realizarem suas tarefas de maneira mais rápida e fácil.

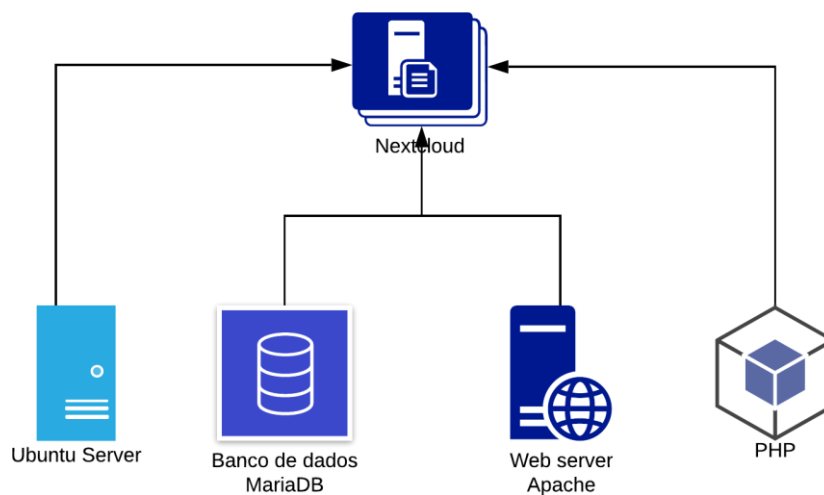
Com a implantação desta solução, será possível integrar as equipes da empresa de uma forma centralizada, totalmente gerenciada pela mesma e que permita que a equipe de TI esteja diretamente envolvida no desenvolvimento e administração da aplicação, além das futuras demandas que possam haver por parte dos funcionários em termos de funções para a mesma. Portanto, em face ao exposto, passemos para a arquitetura da mesma.

Arquitetura da Aplicação

O nextcloud é uma aplicação distribuída que permite a sincronia de conteúdos em diretórios locais como computadores, tablets e outro para um servidor da mesma. Para que isso ocorra com sucesso, as seguintes dependências são utilizadas:

- Um servidor Linux
- Um banco de dados
- Um web server

Por ser uma aplicação de código aberto, ela dá a opção de self-hosted, ou seja, o próprio usuário pode hospedar seu servidor próprio sem incorrer em custos de licenciamento/royalties para isso. O processo de instalação dele é demasiado simples, variando em complexidade de acordo com a opção escolhida: ele pode ser instalado em um servidor Linux, em um container ou em uma nuvem publica (AWS, Azure) ou uma privada (Openstack). Para entender de melhor maneira o processo de instalação, observe o diagrama abaixo:

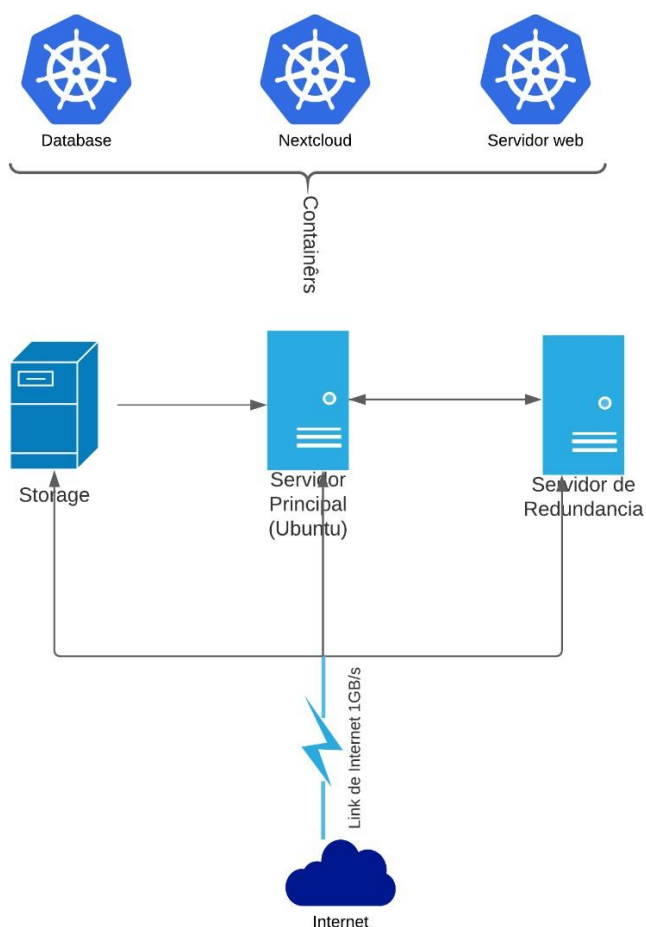


Organização da Infraestrutura

Como a aplicação desejada possui um appliance docker já pronto, bastando realizar sua instalação e posterior configuração, a infraestrutura a ser utilizada para alocação da mesma será montada da seguinte maneira:

- Três containers com:
 - I. um servidor web (Nginx),
 - II. um banco de dados (MariaDB)
 - III. A própria aplicação nextcloud.
- Os containers serão montados em um servidor ubuntu 18.04 com redundância e replicação para outro servidor, garantindo a alta disponibilidade da mesma em caso de falha em um deles.
- Os links de internet a serem utilizados nos servidores serão de 1gb/s, para garantir o mínimo de latência e a máxima velocidade na transferência de arquivos.
- O storage a ser utilizado pela aplicação será de 10TB

O diagrama abaixo explicita de maneira mais clara como será montada a infraestrutura:



Para garantir que a aplicação rodará de maneira satisfatória e ter o desempenho adequado as necessidades organizacionais da empresa, segue-se abaixo as configurações de ambos os servidores, principal e de redundância:

Dell modelo PowerEdge T440 Tower Server com as seguintes especificações:

Processador Intel® Xeon® Gold 5222 3.8G, 4C/8T, 10.4GT/s, 16.5M Cache, Turbo, HT (105W)
DDR4-2933

I. Memória RAM 32GB RDIMM, 2666MT/s, Dual Rank

II. Disco Rígido 1.8TB 10K RPM SAS 12Gbps 512e 2.5in Hot-plug Hard Drive, 3.5in

HYB CARR

Preço: R\$ 19.993,27

O docker é uma aplicação que consiste em uma plataforma de containers, que por sua vez é uma tecnologia que executa partes de softwares de um arquivo completo, proporcionando um ambiente isolado para sua execução e providenciando todos os recursos necessários para o mesmo. A principal vantagem da utilização desta tecnologia se dá pela facilidade de execução proporcionada aos desenvolvedores, que podem executar aplicações sem precisar de realizar configurações complexas para isto, já que o docker possui um vasto repositório de imagens pré-configuradas como o mysql server, ubuntu, nginx dentre outros. Para compreender melhor de que maneira esse processo funciona, segue-se a explicação:

I- O docker destaca recursos e usa bibliotecas em comum para rodar os softwares

II- Ele exclui o virtualizador do processo, ao não necessitar de múltiplos sistemas operacionais para rodar as aplicações. Todas elas são rodadas dentro de um único sistema

III- O docker vem se popularizando como uma alternativa a virtualização por demandar de menos recursos, simplificar a aplicação da metodologia DevOps e facilitar o desenvolvimento ágil, além de possibilitar a execução de deploys e o escalonamento de aplicações.

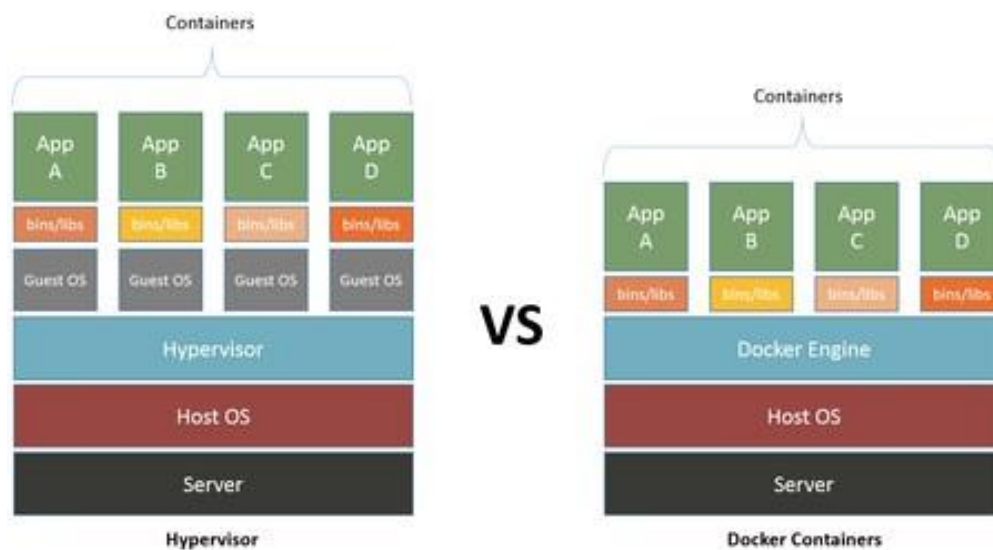
Considerando as informações acima citadas, a tecnologia docker é a mais apropriada para a instalação da aplicação nextcloud, levando em conta sua simplicidade de configuração, o fato de a mesma já possuir uma imagem pré-configurada em seu repositório, e além disso possibilitar que a mesma rode em múltiplos containers, a caracterizando dessa maneira como uma aplicação distribuída.

Virtualização por VMs vs Virtualização por Containers

A diferença principal entre esses dois tipos de virtualização se dá fundamentalmente da forma como funcionam:

A Virtualização por VMs consiste na emulação de um sistema operacional através de um software rodando em cima de servidores físicos para emular um hardware específico. Esse software é chamado hypervisor, é ele que cria e executa as máquinas virtuais. As VMs possibilitam a execução de múltiplos sistemas operacionais sem a necessidade de adquirir servidores adicionais, causando uma redução de custos através da possibilidade de um provisionamento mais rápido e uma capacidade de recuperação de desastres melhorada. A virtualização por VMs diminui custos ao eliminar a atividade de servidores ociosos/com pouca utilização, já que dentro de um mesmo servidor, podem rodar múltiplas aplicações e sistemas operacionais.

A tecnologia de containers por sua vez permite a execução de aplicações rodando apenas o necessário para o seu funcionamento. Ele compartilha o kernel do SO host, os binários e as bibliotecas, porém somente em modo leitura, tornando os containers excepcionalmente leves em comparação com as máquinas virtuais. Para entender melhor essa diferença de funcionamento, segue-se a imagem abaixo:



Como podemos observar, a principal diferença entre esses dois modelos é que os containers não necessitam emular um sistema operacional inteiro para emular uma aplicação. Ao compartilhar recursos com o sistema hospedeiro, torna-se virtualização um processo muito mais simples, leve e que demanda menos esforço administrativo para ser configurado.

Planejamento de implementação da Aplicação

Para realizar a implantação de qualquer aplicação, é necessário um planejamento estruturado, bem organizado, que leve não somente em consideração os custos decorrentes, mas também o impacto que a mesma irá causar no ambiente de trabalho e de que maneira este pode ser reduzido com a adoção de medidas de caráter corretivo. Portanto, face ao exposto, segue-se abaixo o plano elaborado para implantar o nextcloud na empresa Xabaraz Investimentos:

1. Preparar o servidor para a aplicação

Nesta etapa serão realizadas a instalação do sistema operacional hospedeiro da aplicação, o Ubuntu, as dependências que ela precisa para funcionar, como o python e o docker, além de verificar se as especificações técnicas do servidor coincidem com as demandadas pela aplicação.

Prazo Estimado: 1 semana

2. Configuração do script automatizado de instalação

Para automatizar a instalação da aplicação e disponibiliza-la para uso futuro em outros projetos da empresa, um script ansible será utilizado para instalar e configurar os containers da mesma. Por se tratar de uma aplicação distribuída, serão utilizados três containers: banco de dados (Mariadb), servidor web (Nginx) e a aplicação propriamente dita (Nextcloud).

Prazo Estimado: 4 dias

3. Instalação e configuração da aplicação

Nesta etapa será realizado o processo de instalação do nextcloud, a configuração da sua configuração de rede, o apontamento para os banco de dados da empresa, além de realizar testes iniciais em um ambiente separado do de produção para saber se ela está devidamente estável. Aqui também será configurado a persistência dos containers, para que caso ocorra alguma falha ou seja necessário reiniciar o servidor, seja para atualização de seu software ou quaisquer outros motivos, o conteúdo da aplicação não se perca.

Prazo Estimado: 3 semanas

4. Realizar a migração dos arquivos para nextcloud

Com o script pronto e a aplicação instalada, será realizado a sincronização de toda a base de dados da empresa com a mesma. Para evitar quaisquer tipos de problemas com os colaboradores, a aplicação só será permitida para uso geral depois que passar pelos testes de: estabilidade, funcionalidade (se tudo o que o programa de propõe a realizar está funcionando corretamente), performance e segurança. Esta é uma das etapas mais demorada pois é nela que serão realizados todos os testes pela equipe de infraestrutura e de desenvolvimento com a aplicação, para atestar que ela está pronta para a utilização por parte do usuário final.

Prazo Estimado: 2 semanas

5. Liberação da aplicação para uso geral

Nesta última etapa será realizada de maneira gradual a transição para a nova aplicação. Para diminuir os impactos desta mudança, um cronograma será montado visando estruturar os intervalos de tempo em que cada equipe a terá habilitada. A ideia é que em cada semana do mês a aplicação seja liberada para uma equipe, até que toda a empresa esteja usando.

Prazo Estimado: 1 mês

Considerações Finais

Por se tratar de uma empresa de investimentos que lida diretamente com capital de investidores e realiza operações sensíveis diariamente, qualquer transição/mudança deve ser planejada meticulosamente para resguardar o patrimônio da empresa de quaisquer danos. Portanto, a adoção do nextcloud como sistema central da corporação será realizada de forma gradual para habituar os usuários mais organicamente e evitar uma abrupta mudança que pode gerar dificuldades de aprendizado.

Ao final desse processo, todos os funcionários estarão utilizando uma única solução de software para compartilhamento de arquivos, chamadas de voz/vídeo, calendários digitais e chats em grupo. A substituição de múltiplos aplicativos com diferentes propósitos por uma solução all-in-one facilitará não só a interação dos usuários, como também aumentará a produtividade dos mesmos já que não terão mais que se preocupar em escolher qual aplicação utilizar para determinados fins.

TESTE DE PERFORMANCE 8

Referência de versões

Na presente instalação da aplicação nextcloud foram utilizadas diferentes versões de componentes para fazer com que a mesma funcionasse de forma completa. Foram utilizados:

- I. Um servidor Linux ubuntu 18.04
- II. A aplicação ansible
- III. A aplicação Docker
- IV. Um container rodando banco de dados MariaDB
- V. Um container rodando a própria aplicação, nextcloud

O conjunto presente acima pode ser encontrado nos seguintes links abaixo (o ultimo refere-se a instalação como um conjunto para o funcionamento do nextcloud).

Ubuntu: <https://releases.ubuntu.com/18.04/>

Ansible: <https://github.com/ansible/ansible>

Docker: <https://www.docker.com/products/docker-desktop>

MariaDB: <https://mariadb.org/download/>

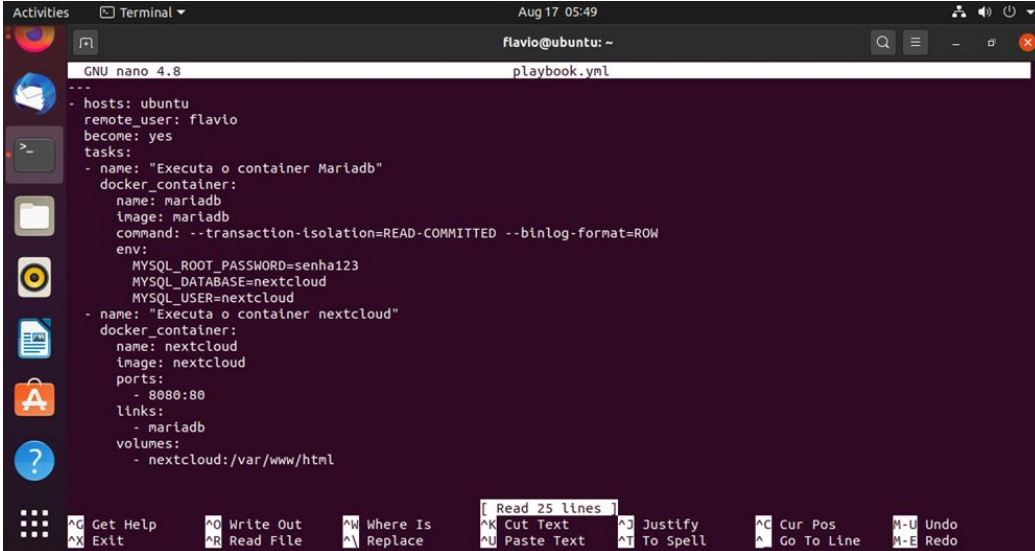
Nextcloud: https://hub.docker.com/_/nextcloud/

Deploy da aplicação

Para instalarmos o nextcloud em formato de container, bem como fazê-lo funcionar de forma persistente para que caso houvesse reinício da máquina a aplicação não se destruísse, foram necessárias algumas configurações adicionais. Os seguintes passos foram realizados:

- I. Deploy de uma VM ubuntu 18.04
- II. Instalação do ansible
- III. Instalação do Docker
- IV. Criação do playbook de instalação da aplicação

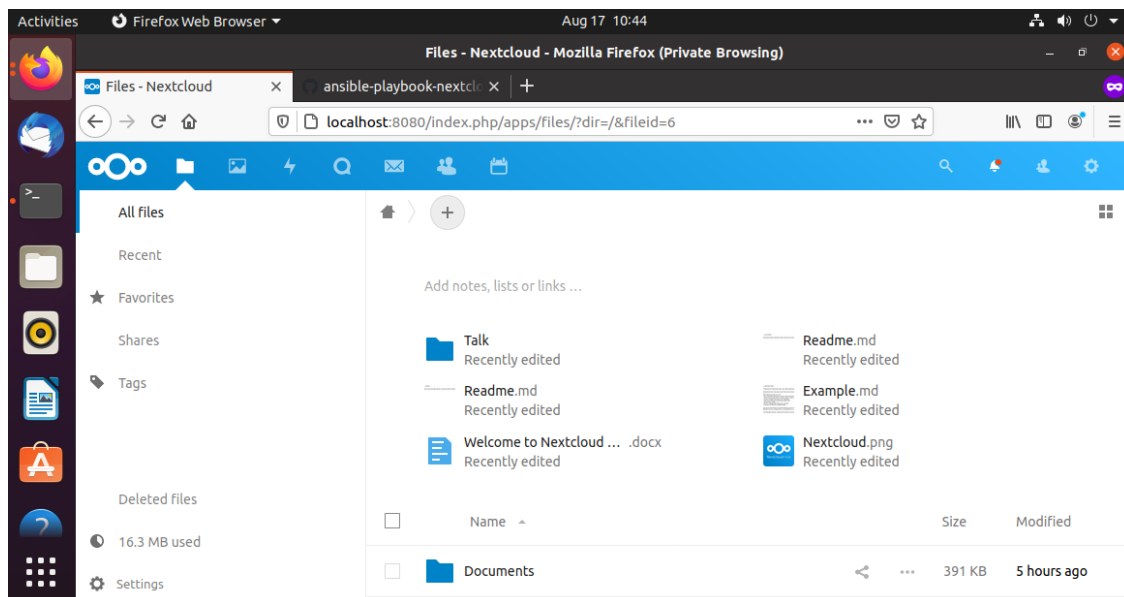
Abaixo seguem-se os prints da configuração do playbook bem como o mesmo rodando dentro da VM:



```
GNU nano 4.8          playbook.yml
---
- hosts: ubuntu
  remote_user: flavio
  become: yes
  tasks:
    - name: "Executa o container Mariadb"
      docker_container:
        name: mariadb
        image: mariadb
        command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
        env:
          MYSQL_ROOT_PASSWORD=senha123
          MYSQL_DATABASE=nextcloud
          MYSQL_USER=nextcloud
    - name: "Executa o container nextcloud"
      docker_container:
        name: nextcloud
        image: nextcloud
        ports:
          - 8080:80
        links:
          - mariadb
        volumes:
          - nextcloud:/var/www/html
```

```
flavio@ubuntu: ~  
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS  
72b3d8d6603e   mariadb    "docker-entrypoint.s..." About an hour ago Up About an hour 3306/tcp  
flavio@ubuntu:~$ ansible-playbook playbook.yml -i hosts.yml -K  
BECOME password:  
PLAY [ubuntu] *****  
TASK [Gathering Facts] *****  
ok: [192.168.159.132]  
TASK [Executa o container Mariadb] *****  
ok: [192.168.159.132]  
TASK [Executa o container nextcloud] *****  
changed: [192.168.159.132]  
PLAY RECAP *****  
192.168.159.132 : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0  
flavio@ubuntu:~$ sudo docker ps  
[sudo] password for flavio:  
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS  
b0e1f1827117   nextcloud  "/entrypoint.sh apac..." 22 minutes ago Up 22 minutes 0.0.0.0:8080->  
80/tcp  
72b3d8d6603e   mariadb    "docker-entrypoint.s..." 2 hours ago   Up 2 hours    3306/tcp  
flavio@ubuntu:~$
```

Reparem que após executar o playbook mencionado, já podemos perceber os dois containers rodando dentro do Docker: mariadb e nextcloud. Para acessar a aplicação, basta colocar o endereço ip da mesma no navegador. Comprovando seu funcionamento, segue-se o print:



CONCLUSÕES

A respeito do prazo de execução

Anteriormente, logo no início do projeto, foi estimado um prazo máximo de 1 mês para implementação completa do ambiente nextcloud na empresa. Por conta de alguns problemas relacionados a adaptação dos usuários ao novo sistema de armazenamento de arquivos, a conclusão do projeto teve de se estender por mais duas semanas. Foi o prazo necessário para ensinar os usuários a interagir com a aplicação, se comunicar através dela, armazenar seus arquivos, e colocá-la como sua principal em seu cotidiano de trabalho. Houveram algumas resistências, claro, mas dentro de um processo de mudança de aplicação para um cenário all-in-one, é de se esperar que alguns tenham mais dificuldade de se desapegar a um conjunto específico de ferramentas com as quais tem trabalhado durante muitos anos. Apesar disso, ao final das duas semanas todo o contingente de funcionários se adaptou perfeitamente a ferramenta e agora temos todos os setores integrados através dela.

A respeito do uso dos recursos

O servidor montado para comportar a aplicação nextcloud foi pensado em um cenário de possível expansão da empresa, já se antevendo a quaisquer tipos de problema que poderiam ser ocasionados caso o servidor fosse dimensionado erroneamente. Com a infraestrutura montada, foi possível atender perfeitamente a todos os usuários, sem travamentos ou qualquer tipo de problemas relacionado tanto ao armazenamento de arquivos quanto aos aplicativos de comunicação da solução. A quantidade de memória ram (32GB) aliadas a um processador bastante potente (Intel® Xeon® Gold 5222) foi possível extrair o máximo de performance sem impactar na vida útil do equipamento.

A respeito das funções da aplicação

O nextcloud foi pensado como uma alternativa all-in-one para substituir a dezena de diferentes aplicações que os usuários utilizavam para comunicar-se entre si e armazenarem seus arquivos. Como uma solução de nuvem, ela cumpriu perfeitamente com seu propósito, principalmente pelo fato de já possuir aplicativos próprios para android e ios, o que facilitou muito a migração dos usuários para o novo sistema. Apenas a parte de comunicação, o nextcloud talk, que ainda está sendo adotado aos poucos. Muitos funcionários tem como principal ferramenta de comunicação o Skype, e relutam para aprender a mexer e realizar suas tarefas em uma nova aplicação. Aos poucos, com a adesão cada vez maior dos líderes de equipe a ferramenta, outros funcionários vão se sentindo mais confortáveis e começam a solicitar a equipe de TI auxílio para ensiná-los a trabalhar com ela.

Sobre o deploy da ferramenta em nuvem

No projeto proposto, estamos aproveitando a infraestrutura on premise da empresa para instalar a aplicação pretendida. Porém, partindo do princípio que todo o deploy da ferramenta está sendo realizado através de máquina virtual e containers, iremos conjecturar como seria sua implementação em uma nuvem privada, mais especificamente, a VMware Cloud, que utiliza o próprio ambiente VMware para montar sua nuvem. Nos parágrafos a seguir será descrito como cada componente da aplicação funcionaria dentro da nuvem privada, bem como quais tecnologias seriam empregadas para realizar seu funcionamento.

Máquina Virtual

Nossa aplicação utiliza uma VM ubuntu como sistema operacional base para funcionar. Ela seria desenvolvida na instância ESXI do VMware, o seu virtualizador. Ele atua como um hipervisor, para provisionar máquinas virtuais, agindo na integração dos componentes do sistema operacional, como o núcleo.

Armazenamento

A ideia principal por trás da adoção do nextcloud é sua utilização para armazenamento de arquivos em nuvem. Para isso, adotando uma solução VMware Cloud, utilizaremos o componente vSAN. O vSAN é um software de virtualização de armazenamento que permite gerenciar o processamento e o armazenamento em uma única plataforma. Com ela, poderemos criar um cluster de HDs a ser utilizado pela aplicação, oferecendo redundância, backup de arquivos e uma solução integrada ao ambiente.

Gerenciamento da infraestrutura

O vCenter será utilizado para obter uma visão centralizada de toda a infraestrutura da nuvem. Com ele será possível provisionar VMs, gateways virtuais, alocar mais recursos computacionais para o sistema operacional e por fim, apresentar uma interface de fácil aprendizado para uso. É por ele que será gerenciado a VM ubuntu, verificar a quantidade

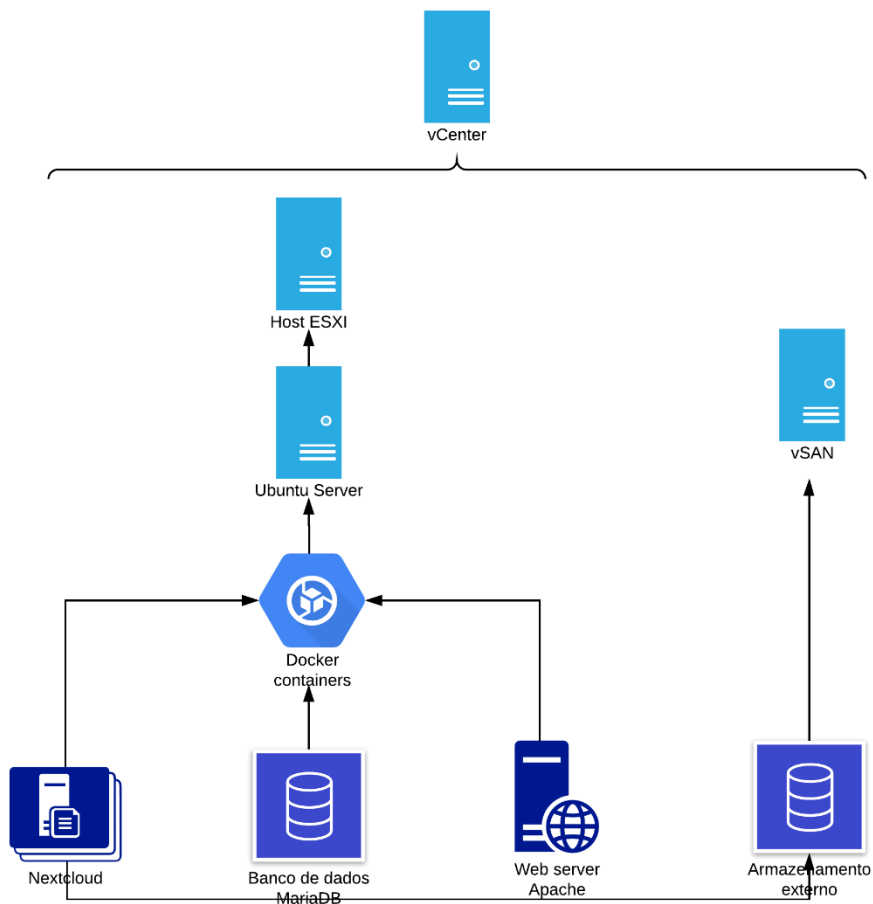
de recursos que está utilizando, se está sobrecarregada ou não, e todas as estatísticas relacionadas a nossa nuvem privada.

Banco de dados

O banco de dados seria instalado dentro da VM ubuntu, através do script ansible mencionado neste documento. O banco a ser utilizado será o MariaDB, gratuito e totalmente integrado ao nextcloud.

Diagrama da configuração

Para compreender melhor de que maneira a aplicação se comportará dentro do ambiente da VMware Cloud, segue-se o diagrama abaixo:



Algumas considerações:

- I. Os componentes da aplicação (banco de dados, servidor nginx) serão instalados e configurados dentro de containers, cada um tendo o seu próprio
- II. Dentro da VM ubuntu a aplicação rodará através dos containers, que serão configurados em modo de persistência para que não se apaguem em caso de reinício do servidor
- III. O nextcloud se utilizará de um armazenamento externo configurado no vSAN para armazenar todos os arquivos dos usuários.
- IV. Toda a infraestrutura (VM, vSAN, interface de rede) será monitorada e gerenciada através do vCenter

A aplicação ficará disponível aos funcionários da empresa dentro e fora das dependências dela. Em caso de acesso externo, a equipe de TI irá configurar uma VPN para que o usuário acesse a rede interna da empresa e possa se conectar remotamente com a aplicação, seja por seu laptop ou smarthphone.

FONTES

<https://portal.nextcloud.com/article/nextcloud-talk-high-performance-backend-requirements-46.html>

https://www.dell.com/en-us/work/shop/servers-storage-and-networking/poweredget440-tower-server/spd/poweredget440/pe_t440_12421c_vi_vp?configurationid=690f24fe-996c-40dd-9ea8-8b4be0e015db

https://hub.docker.com/_/nextcloud/

<https://github.com/nextcloud/docker>