



INSTITUTO INFNET
ESCOLA SUPERIOR EM TECNOLOGIA DA INFORMAÇÃO - ESTI
GRADUAÇÃO EM GESTÃO DE TECNOLOGIA DA INFORMAÇÃO – GGTI
PROJETO DE BLOCO ARQUITETURA E INFRAESTRUTURA DE APLICAÇÕES
DIAFANNYS ALBUQUERQUE SAMPAIO

PROJETO DE IMPLANTAÇÃO DE APLICAÇÃO DISTRIBUIDA –
PROJETO DE BLOCO

MATRICULA 09306896409

RIO DE JANEIRO

2017

SUMÁRIO

1. INTRODUÇÃO	3
2. OBJETIVO	3
3. JUSTIFICATIVA	3
4. INFORTECNOSEG	4
5. PROJETO DE IMPLANTAÇÃO ODOO	4
6. IMPLANTAÇÃO DE SERVIDORES	6
6.1 ESXI	7
6.2 VCENTER	9
6.3 AMAZON WEB SERVICES	12
6.3.1 IAM – INDENTITY AND ACESS MANAGEMENT	13
6.3.2 ROUTE 53	14
6.3.3 VPC – VIRTUAL PRIVATE CLOUD	14
6.3.4 EC2 – ELASTIC COMPUTING CLOUD	14
6.3.5 AMAZON S3	14
6.3.6 RDS - AMAZON RELATIONAL DATABASE SERVICE	14
7. CUSTOS DO PROJETO	15
7.1 CAPEX	15
7.2 OPEX	15
8. PREPARANDO O AMBIENTE PARA A APLICAÇÃO	16
8.1 OPENSSEH-SERVER	16
8.2 VIM	17
9. AUTOMAÇÃO	17
9.1 ANSIBLE	17
9.2 DOCKER	18
9.3 GIT	20
10. IMPLANTAÇÃO DA APLICAÇÃO	20
11. CONCLUSÃO	31
12. BIBLIOGRAFIA	33

1. INTRODUÇÃO

Teste de performance com foco na implantação de aplicação em arquitetura em nuvem, dentro do Projeto de Bloco Arquitetura e Infraestrutura de Aplicações.

2. OBJETIVO

Esse trabalho tem por objetivo a avaliação de desempenho referente ao projeto de Implantação de aplicação em Infraestrutura em Nuvem. Nesse projeto, será demonstrado um escopo do projeto de implantação de uma aplicação distribuída, através de implantação de servidores, Virtual Machines e infraestrutura de nuvem, dentre elas a nuvem pública, com a implantação de VPC e criação de instancias, além de demonstração de acesso das instancias via protocolo SSH e da aplicação via web, descritos no decorrer do teste de performance. A base para a criação dessa infraestrutura, foram apresentados em sala de aula e através de material disponível na plataforma Moodle e através de pesquisas pela internet como descrito na bibliografia.

3. JUSTIFICATIVA

Este trabalho tem por foco testar as habilidades do aluno em relação ao conteúdo apresentados do corpo do Projeto de Bloco Arquitetura e Infraestrutura de Aplicações.

4. INFORTECNOSEG

Para uma empresa ter um crescimento impactante e visibilidade no mercado, é necessário a inovação e aprimoramento constante de seus serviços, com essa definição, damos início a apresentação da empresa infortecnoseg, empresa da área de Tecnologia da informação com atuação em distintos seguimentos da área de Tecnologia. Buscando a expansão de mercado e uma maior visibilidade, a empresa vem passando por algumas modificações acompanhando a evolução do setor de serviços e com a busca de atender o mercado com as novas tecnologias. Inicialmente, a empresa com sede no Rio de Janeiro, tinha como foco o setor de suporte de TI e sistemas de segurança, no qual ainda possui clientes nesse seguimento. Com a evolução dos serviços, a empresa está buscando aprimoramento e novos seguimentos, e com isso se adequando a expansão do mercado de tecnologia. Com isso, a ampliação da empresa acaba tornando necessário a ampliação das aplicações oferecidas pela empresa e a gestão de seus serviços. O diretor da empresa, optou pela implantação de uma nova ferramenta após analisar as opções de aplicações que tivesse como atender as necessidades e fosse possível adaptar para gestão e serviços da empresa, com esse objetivo, escolheu a aplicação distribuída Odoo para otimizar os serviços oferecidos pela organização e otimizar a gestão da empresa, e com isso o provisionamento de novos serviços para seus clientes.

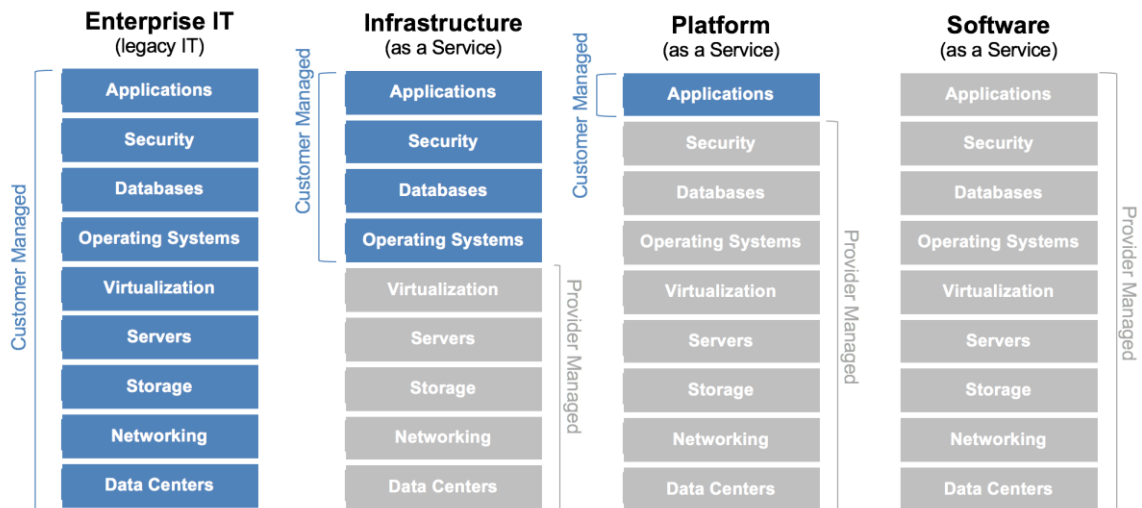
Foi feito um estudo para verificar a necessidade para expansão dos serviços e implantação dessa nova aplicação distribuída, e com isso foi levantado um estudo de melhor custo benefício para empresa, levando em conta a disponibilidade, a latência, valor de manutenção e mão de obra qualificada, além do custo de infraestrutura para a implantação dessa nova aplicação.

5. PROJETO DE IMPLANTAÇÃO ODOO

O projeto de implantação da aplicação distribuída Odoo, foi escolhida após um estudo de benefícios e em relação aos custos para implantação dentro da empresa Infortecnoseg, estudo esse levantado pelo departamento de tecnologia da informação. Foi levado em conta para a escolha, o expressivo momento de cloud Computing, no

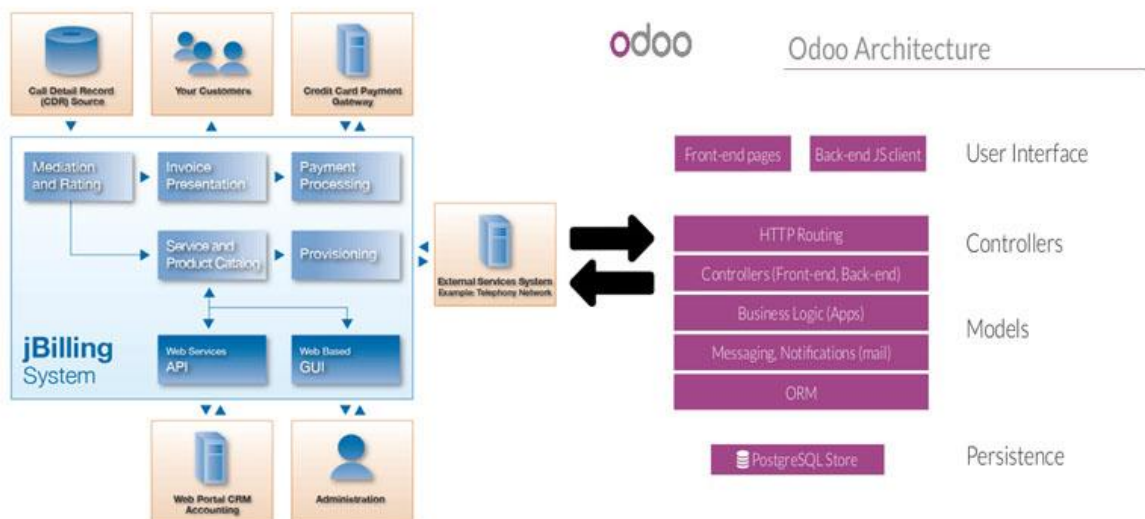
qual oferece alta disponibilidade, além da avaliação de custo benefício para a empresa.

Para a implantação da aplicação, foi levantado a opção de implantação de nuvem híbrida, com a utilização da AWS, levando em conta a alta disponibilidade e o custo benefício. E também a implantação do serviço com a utilização de servidores com a plataforma da VMware. Para a implantação, foi selecionado a infraestrutura de serviço com a implantação de um servidor físico com o sistema operacional Linux Ubuntu, e a utilização do Vsphere da vmware como plataforma de computação em nuvem privada, que é uma distribuição Open Source, projetada como um ambiente de infraestrutura como serviço (IaaS). Além do Vmware, foi selecionado a utilização da Amazon Web Services como uma forma de alta disponibilidade para o acesso da aplicação. Esses serviços poderiam ser substituídos por algumas outras alternativas, como exemplo a Azure da Microsoft, como infraestrutura de serviços em nuvem, além de inúmeros outras empresas que oferecem os serviços de Cloud Computing. O que acabou sendo selecionadas as tecnologias da VMware, e AWS após um comparativo, levando em conta o melhor custo benefício, para aplicação do projeto.





A aplicação Odoo é uma solução de Gestão empresarial Open Source, que tem como destaque aplicações voltadas para áreas de contabilidade, gerenciamento de materiais, gestão de venda e compras, gestão de recursos humanos, marketing, help desk dentre outros. O sistema é modular e flexível, o que proporciona uma maior autonomia para escolha dos módulos que são necessários para implantação na organização de acordo com as necessidades e projeto.



O Odoo possui uma estrutura em três camadas principais, que é composta por Servidor, cliente e banco de dados. O banco de dados utilizado é o PostgreSQL, já o servidor é escrito em Python e é modular, com estrutura pré-definida. Por fim, utilizamos o cliente via interface Web.

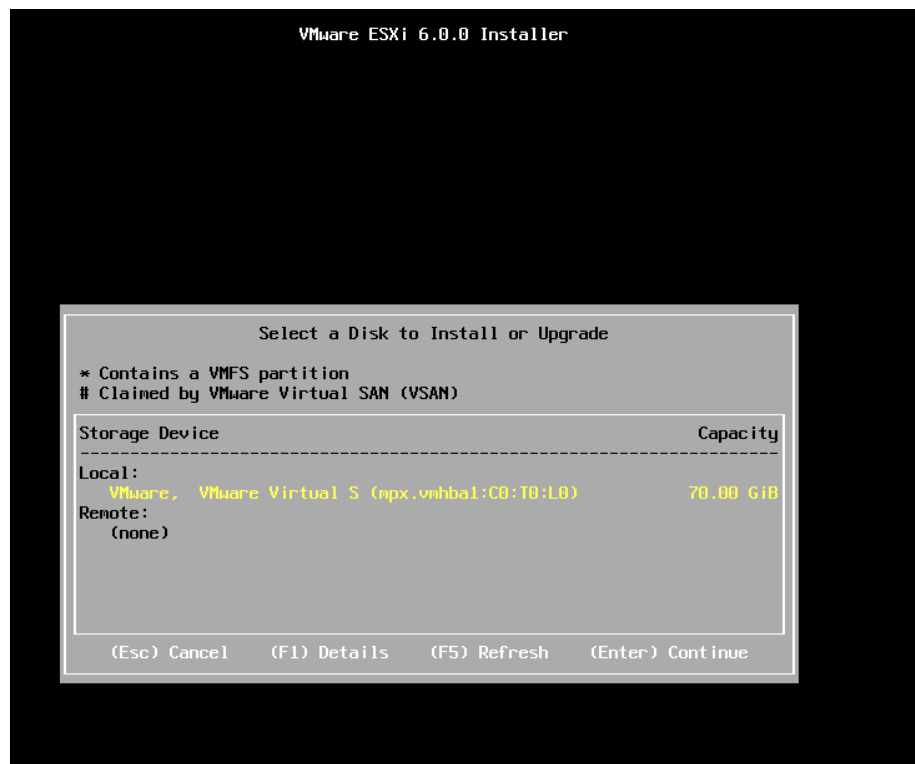
6. IMPLANTAÇÃO DE SERVIDORES

Na fase de implantação do projeto, damos início a implantação dos servidores e aplicações necessárias para o projeto.

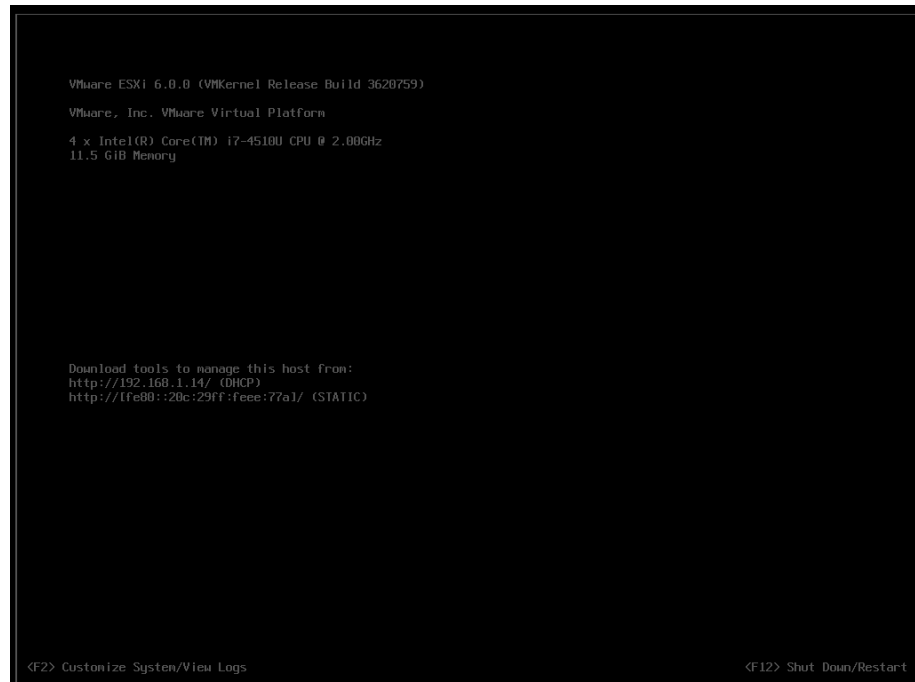
6.1 ESXi

O Servidor R530 será utilizado para instalação do Hipervisor ESXi da VMware, que orienta para instalação do ESXi um cpu de 2 cores 64bits, mínimo de 4gb de memória física, no entanto, é recomendado a utilização de 32gb de memória RAM para rodar as Vms.

Dando sequência, é demonstrado a instalação do ESXi no qual será implantado as Vms do projeto.





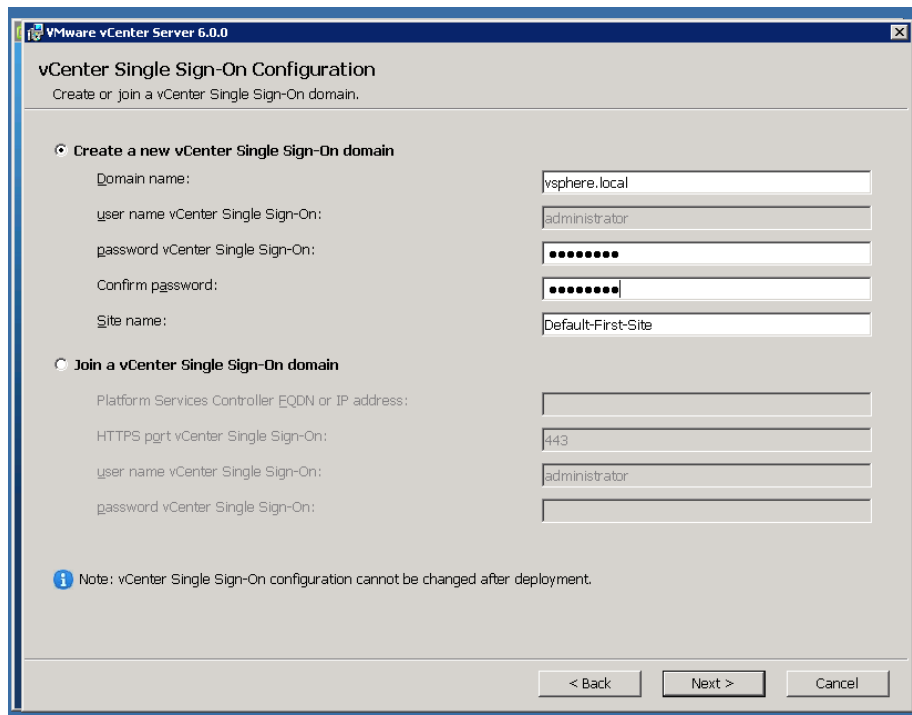
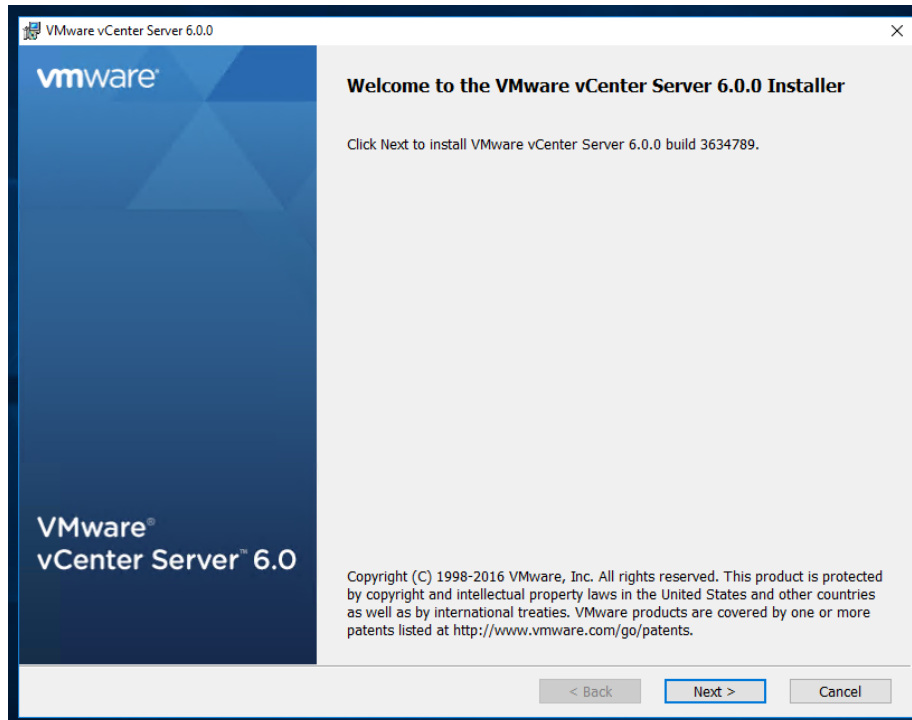


Após a instalação é iniciado a configuração do ESXi, para a configuração do datacenter, e por fim é instalado o Vcenter para gerenciamento do Datacenter.

6.2 VCENTER

Dando continuidade, vamos dar início a implantação do Vcenter dentro do Windows Server 2016, com mínimo de 2cpu 64bits, e recomenda a utilização de 8Gb de memória para um bom desempenho. Em seguida, podemos ver o processo de instalação do Vcenter.

É uma instalação que merece atenção em relação principalmente na configuração do domínio, vale lembrar, que o domínio não é o domínio da empresa, e sim um domínio do datacenter.




VMware vCenter Server 6.0.0

Configure Ports

Configure network settings and ports for this deployment.

Common Ports	
HTTP Port:	80
HTTPS Port:	443
Syslog Service Port:	514
Syslog Service TLS Port:	1514
Platform Services Controller Ports	
Secure Token Service Port:	7444
vCenter Server Ports	
Auto Deploy Management Port:	6502
Auto Deploy Service Port:	6501
ESXi Dump Collector Port:	6500
ESXi Heartbeat Port:	902
vSphere Web Client Port:	9443

 The following ports must also be available for this deployment:
25, 88, 111, 389, 636, 832, 2012, 2014, 2020, 5480, 7080, 7081, 8200, 8300, 8301, 11711, and 11712

< Back Next > Cancel

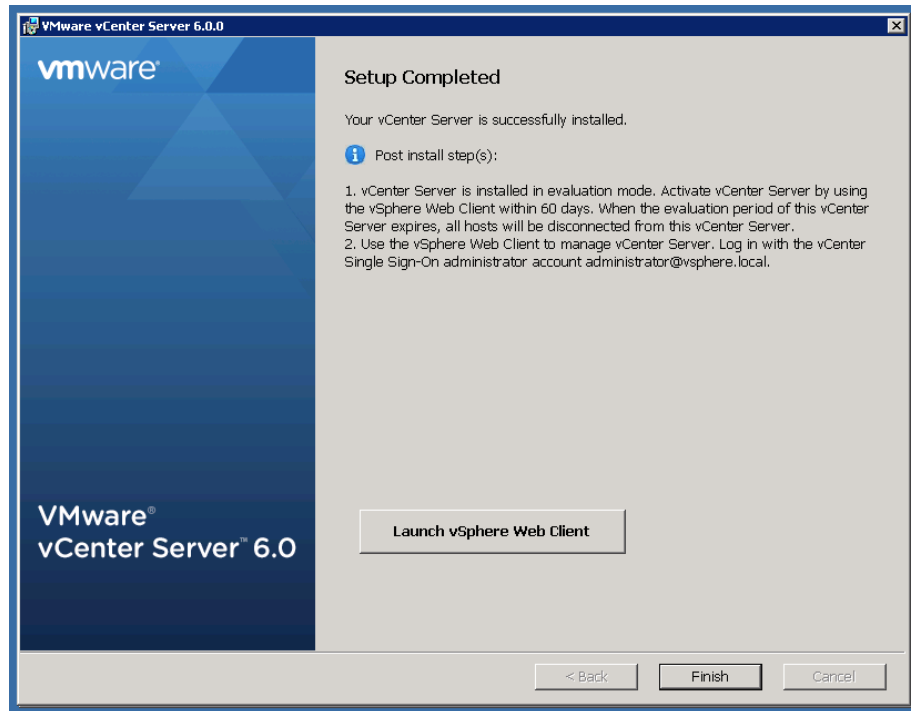
VMware vCenter Server 6.0.0

Ready to install

Review your settings before starting the installation.

System Name:	192.168.0.11
Deployment type:	vCenter Server with an embedded Platform Services Controller
vCenter Single Sign-On configuration:	Create a new vCenter Single Sign-On domain
vCenter Single Sign-On user name:	administrator
vCenter Single Sign-On domain:	vsphere.local
vCenter Single Sign-On site name:	Default-First-Site
vCenter Server service account:	Windows Local System Account
Database type:	embedded (vPostgres)
Installation directory:	C:\Program Files\VMware\
Data directory:	C:\ProgramData\VMware\

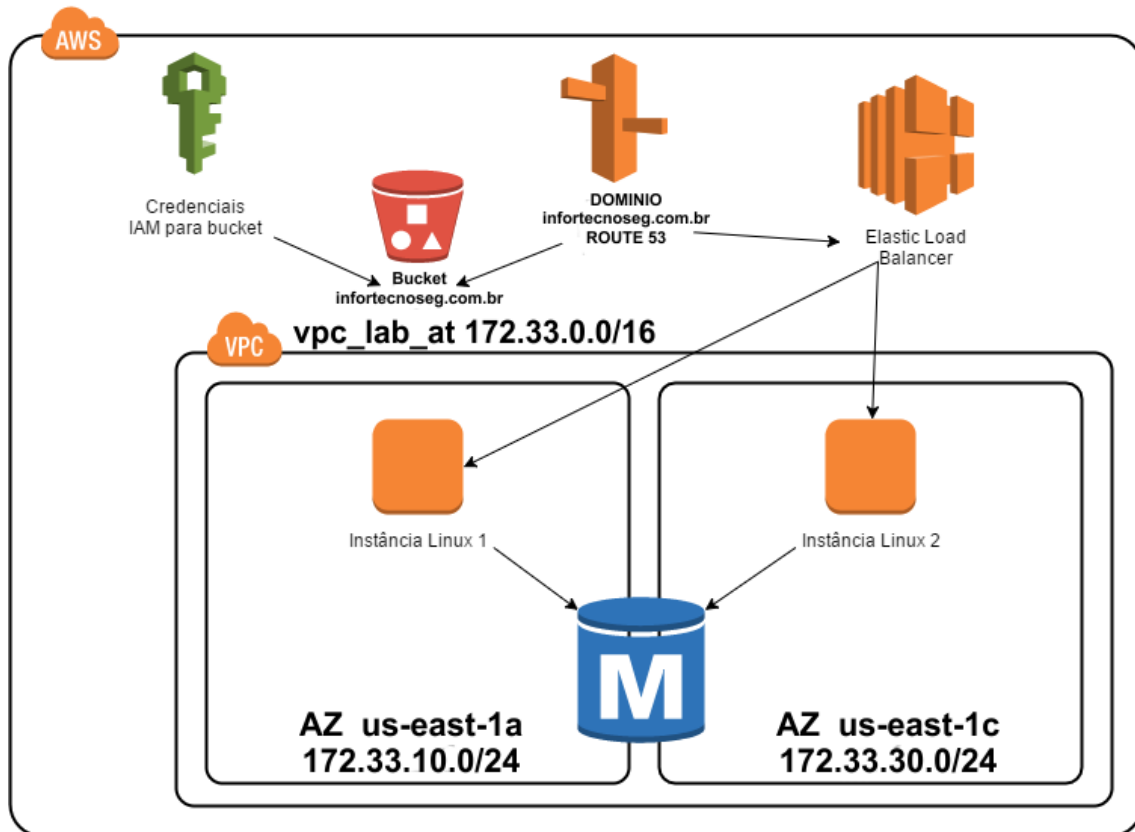
< Back Install Cancel



Após o fim da instalação, é iniciado o processo de configuração do datacenter, atendendo as necessidades do Datacenter. Na sequência do projeto, serão destacados os pontos principais que necessita de um cuidado especial e principalmente atendendo as necessidades para isolar as redes internas e publicas dentro do Datacenter em questão.

6.3 Amazon Web Services

Dando continuidade ao projeto, será Implantação a Infraestrutura em Nuvem Pública para esse projeto, como foi definido pelo departamento de tecnologia da informação. A Amazon Web Services, será aplicada a infraestrutura da infortecnoseg, aonde será balanceado o acesso de serviço web dentro de duas áreas de disponibilidades diferentes, dentro de uma vpc configurada para acesso via web pelo site infortecnoseg.com.br, e com isso, serão implantados duas instancias com Linux ubuntu aonde será implantado o servidor com a aplicação.



Para esse projeto, serão configurados os serviços IAM - Identity and Access Management, a criação de um Bucket com o serviço S3, aonde será armazenada a pasta raiz do site infortecnoseg.com.br, além disso, com o serviço ROUTE 53, será implantado uma zona de DNS, e utilizando o serviço VPC – Virtual Private Cloud, aonde será configurada a nuvem pública, aonde serão hospedados os serviços dentro da amazon, e por fim, a criação de duas instancias e um banco de dados distribuídos dentro do vpc nas zonas de disponibilidade dentro da região escolhida na AWS, esses serviços serão detalhados nos tópicos a seguir.

6.3.1 IAM – INDENTITY AND ACESS MANAGEMENT

Para controle de acesso e identidade, a Amazon utiliza o IAM – Identity and Acess Management, com isso, é possível a criação de grupos e usuários, que será atribuindo a cada usuário um nível de acesso aos serviços e projetos dentro da AWS, essa definição fica por conta do usuário máster da conta aonde serão hospedados os serviços dentro da Amazon Web Services.

6.3.2 ROUTE 53

O provisionamento e controle de Domínios e sistemas de nome de domínio (DNS), são administrados dentro da plataforma Route 53, no qual são direcionados os endereços ip das aplicações e serviços e traduzidos para nomes.

6.3.3 VPC – VIRTUAL PRIVATE CLOUD

A Amazon Virtual Private Cloud (Amazon VPC), como sugere o nome, é um serviço de definição de cloud Computing, através da VPC, no qual podemos isolar ou não um ambiente, com isso podemos fazer de acordo com cada projeto a definição do que está acessível ao público ou de modo privado e isolado para acesso dentro da organização, por fim, pode ser otimizado a segurança das instancias e serviços dentro da AWS, de acordo com cada projeto.

6.3.4 EC2 - ELASTIC COMPUTE CLOUD

O serviço EC2 é a parte computacional da Amazon, com esse serviço, podemos administrar a criação de instancias redimensionável na nuvem, através desse serviço, podemos de forma dinâmica e escalável no provisionamento de computação em nuvem.

6.3.5 AMAZON S3

O Amazon S3, é um serviço de armazenamento dentro da AWS, reponsavel pelo armazenamento de Objetos dentro da nuvem. Com isso, podemos ter de modo econômico e preciso a alta disponibilidade de objetos dentro da Amazon S3.

6.3.6 RDS - AMAZON RELATIONAL DATABASE SERVICE

Serviço responsável pela administração e operação de Bancos de dados de forma simples e automatizadas, além de ser um serviço dinâmico, de alta disponibilidade e alto desempenho.

7 CUSTOS DO PROJETO

Para execução desse projeto, foi levantado os custos com compra de servidores, licenças da Vmware, e custos mensais com a Amazon Web Services, e custos com pagamento de funcionários para execução do projeto.

7.1 Capex

As despesas de investimento, serão aplicadas com fundos destinados ao projeto. Para execução do projeto em destaque, é necessário a aquisição de servidores e licenças para operação além da infraestrutura de rede, como descritos na tabela em anexo.

Equipamento / Licenças	Preço da licença
VMware vSphere Enterprise Plus	US\$ 3.495,00
Dell PowerEdge R530	R\$ 15.749,00
Infraestrutura (Rede, rack, mobiliários e etc...)	R\$ 25.000,00

7.2 Opex

Os custos operacionais, serão mantidos pela empresa, com isso é necessário adequar o quadro de funcionários com as funções necessárias para execução do projeto, além de possível contratação para execução de serviços implantados com o novo projeto.

DevOps	Fundo Mensal
Desenvolvimento	R\$ 10.495,00
Operações	R\$10.749,00

Para esse projeto, foi feito levantamento de custos fixos com infraestrutura e operacional. Para ampliação dos serviços, será necessário a contratação dos serviços da **Amazon Web Services**, valores que serão utilizados mediante as necessidades de expansão do projeto.

PLANO DE IMPLANTAÇÃO		
SERVIÇO	Prazo	Departamento Responsável
Planejamento	20/01/2017	Departamento de TI
Orçamento	10/02/2017	Departamento de Compras
Compra de equipamentos e licenças	05/03/2017	Departamento de Compras
Instalação da Infraestrutura	10/04/2017	Departamento de TI
Implantação da Nuvem pública	05/05/2017	Departamento de TI
Implantação da Nuvem Privada	10/05/2017	Departamento de TI
Implantação da aplicação	15/05/2017	Departamento de TI

8 PREPARANDO O AMBIENTE PARA A APLICAÇÃO

Para a implantação da aplicação, será necessário a preparação do ambiente no qual será instalada a aplicação. Como descrito no projeto, a aplicação será implantada dentro do sistema operacional Ubuntu 16.04.

Após a instalação do sistema operacional, será necessária, configurações adicionais para preparar o ambiente para receber a aplicação. Primeiro, é importante a atualização dos pacotes do sistema, para isso são utilizados os comandos a seguir;

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

8.1 Openssh-server

Para a instalação do openssh-server, é necessário seguir os comandos passo a passo para configurar o acesso via ssh.

\$ sudo apt-get install openssh-server (comando para instalação do pacote openssh-server)

\$ ssh-keygen (com esse comando são criadas as chaves pública e privada)

\$ cp -p ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys (com esse comando é copiado a chave pública para a pasta de chaves autorizadas)

8.2 Vim

O pacote vim é um editor de texto muito utilizado por não possuir uma interface gráfica, se torna muito utilizado principalmente para edição de arquivos com acesso pelo terminal. Para instalação, é utilizado o comando

\$ sudo apt-get install vim (comando para instalação do pacote vim)

9 AUTOMAÇÃO

Com o objetivo de otimizar o processo de implantação do serviço, a organização optou por utilizar uma ferramenta de gerenciamento de configuração, no qual foi escolhido a ferramenta Ansible, que utiliza uma linguagem específica de domínio que é utilizada para descrever o estado de suas máquinas. Com isso, utilizaremos essa ferramenta com foco em otimizar o processo de implantação da aplicação escolhida.

9.1 Ansible

Para a automatização com o Ansible, não é necessário instalar o mesmo nas máquinas de destino, já que a ferramenta utiliza o protocolo ssh para envio de comando de automação. Com isso, será necessário a instalação apenas na máquina no qual será responsável para escrita do playbook para automação do serviço.

Para dar início a instalação, é necessário a instalação de um repositório no qual contem a versão mais atualizada do Ansible. Para isso, é utilizado o seguinte comando.

\$ sudo apt-add-repository ppa:ansible/ansible

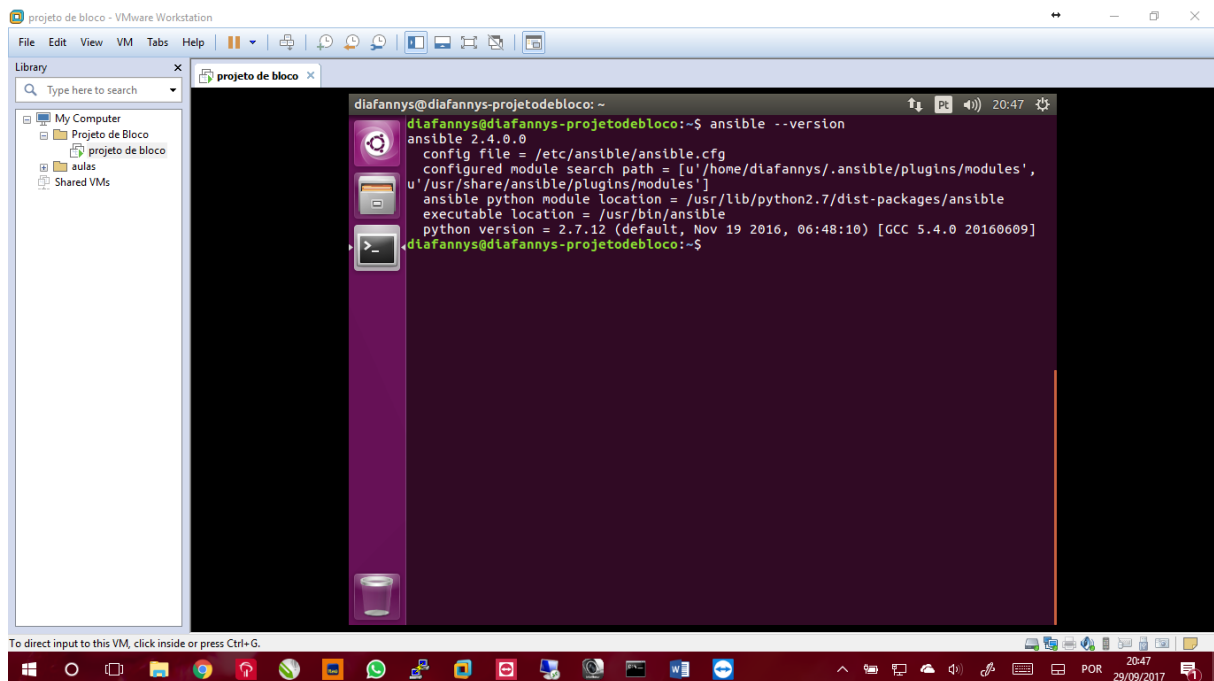
Após a adição do repositório, é necessário a atualização do sistema para obter os pacotes de instalação atualizados.

\$ sudo apt-get update

Com o sistema atualizado, iremos iniciar a instalação do Ansible, e verificar a versão que foi instalada.

\$ sudo apt-get install ansible

\$ ansible --version



```

projeto de bloco - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Projeto de Bloco
projeto de bloco
aulas
Shared VMs
projeto de bloco
diafannys@diafannys-projetodebloco: ~
diafannys@diafannys-projetodebloco:~$ ansible --version
ansible 2.4.0.0
config file = /etc/ansible/ansible.cfg
configured module search path = [u'/home/diafannys/.ansible/plugins/modules',
u'/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python2.7/dist-packages/ansible
executable location = /usr/bin/ansible
python version = 2.7.12 (default, Nov 19 2016, 06:48:10) [GCC 5.4.0 20160609]
diafannys@diafannys-projetodebloco:~$

```

9.2 Docker

O Docker, é uma plataforma de software, que permite a criação, teste e implantação de aplicações de modo rápido, com a utilização de containers, ele cria pacotes de softwares necessários para implantação de uma aplicação. Com a utilização de containers, é possível uma rápida implantação de uma aplicação. Com essa breve definição, iremos implantar o ambiente para execução de containers Docker, para execução de nossa aplicação. Para utilização de containers Docker, será necessário a instalação dessa plataforma. Com estes comandos, de modo rápido, iremos implantar essa plataforma.

**\$ sudo apt-get install **

**apt-transport-https **

**ca-certificates **

**curl **

software-properties-common

Nesse momento, faremos o download da chave de assinatura digital do projeto Docker e adicionaremos ao nosso gerenciador de pacotes

\$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

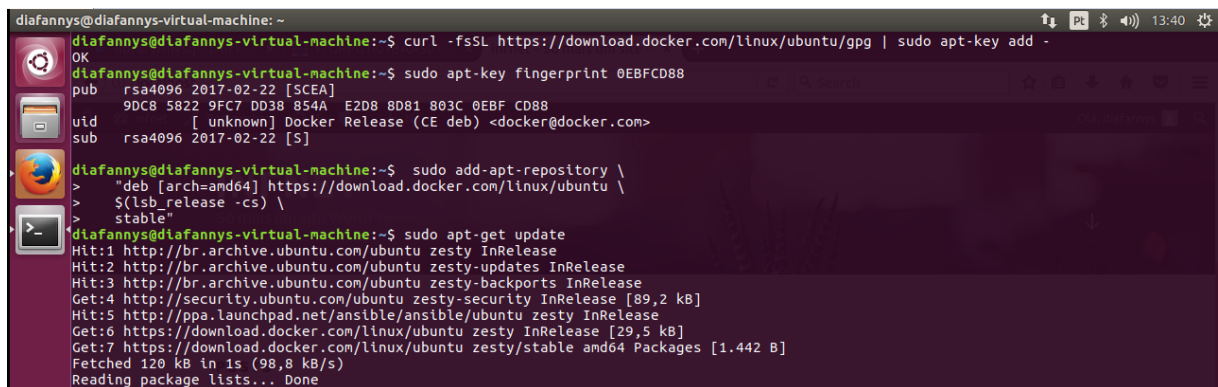
Com esse comando, iremos adicionar o repositório de pacotes do projeto Docker.

```
$ sudo add-apt-repository \  
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) \  
stable"
```

Com o repositório instalado, é necessário a atualização dos pacotes com o comando de atualização de pacotes

\$ sudo apt-get update

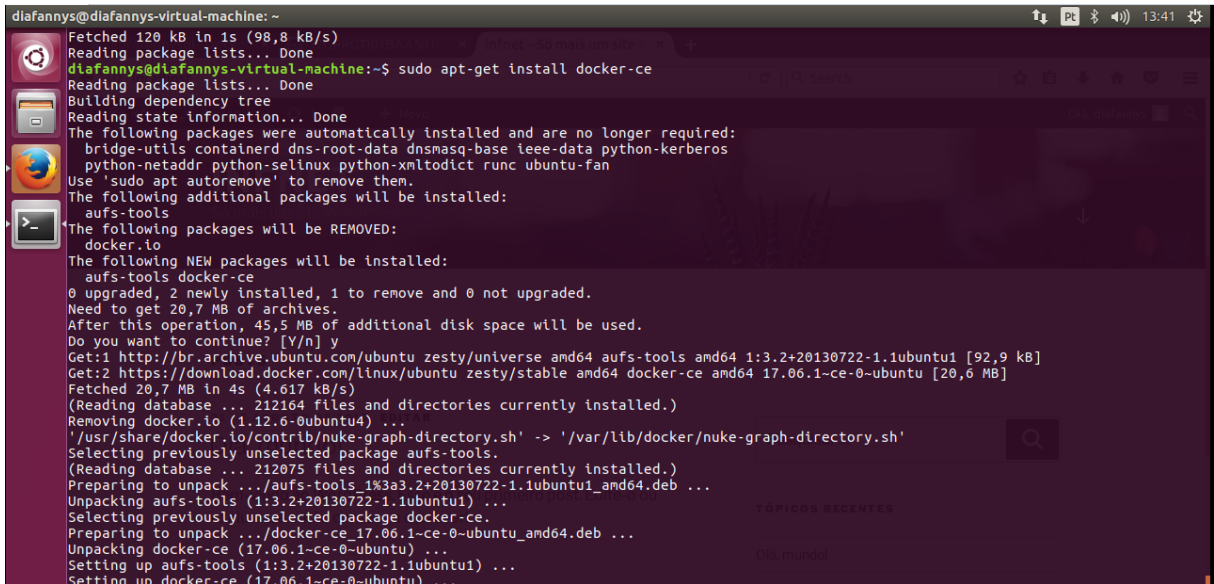
Com os pacotes atualizados, nesse momento iremos instalar o sistema Docker, que será necessário para execução dos containers.



```
diafannys@diafannys-virtual-machine: ~  
diafannys@diafannys-virtual-machine:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
OK  
diafannys@diafannys-virtual-machine:~$ sudo apt-key fingerprint 0EBFCD88  
pub rsa4096 2017-02-22 [SCEA]  
9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88  
uid [ unknown] Docker Release (CE deb) <docker@docker.com>  
sub rsa4096 2017-02-22 [S]  
diafannys@diafannys-virtual-machine:~$ sudo add-apt-repository \  
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
> $(lsb_release -cs) \  
> stable"  
diafannys@diafannys-virtual-machine:~$ sudo apt-get update  
Hit:1 http://br.archive.ubuntu.com/ubuntu zesty InRelease  
Hit:2 http://br.archive.ubuntu.com/ubuntu zesty-updates InRelease  
Hit:3 http://br.archive.ubuntu.com/ubuntu zesty-backports InRelease  
Get:4 http://security.ubuntu.com/ubuntu zesty-security InRelease [89,2 kB]  
Hit:5 http://ppa.launchpad.net/ansible/ansible/ubuntu zesty InRelease  
Get:6 https://download.docker.com/linux/ubuntu zesty InRelease [29,5 kB]  
Get:7 https://download.docker.com/linux/ubuntu zesty/stable amd64 Packages [1.442 B]  
Fetched 120 kB in 1s (98,8 kB/s)  
Reading package lists... Done
```

\$ sudo apt-get install docker-ce

E com isso estamos com nosso sistema Docker em funcionamento, disponível para execução dos containers do nosso projeto.



```

dlafannys@dlafannys-virtual-machine: ~
Fetched 120 kB in 1s (98,8 kB/s)
Reading package lists... Done
dlafannys@dlafannys-virtual-machine:~$ sudo apt-get install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils containerd dns-root-data dnsmasq-base ieee-data python-kberos
  python-netaddr python-selinux python-xlntdict runc ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  aufs-tools
The following packages will be REMOVED:
  docker.io
The following NEW packages will be installed:
  aufs-tools docker-ce
0 upgraded, 2 newly installed, 1 to remove and 0 not upgraded.
Need to get 20,7 MB of archives.
After this operation, 45,5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://br.archive.ubuntu.com/ubuntu zesty/universe amd64 aufs-tools amd64 1:3.2+20130722-1.1ubuntu1 [92,9 kB]
Get:2 https://download.docker.com/linux/ubuntu zesty/stable amd64 docker-ce amd64 17.06.1-ce-0~ubuntu [20,6 MB]
Fetched 20,7 MB in 4s (4.617 kB/s)
(Reading database ... 212164 files and directories currently installed.)
Removing docker.io (1.12.6-0ubuntu4) ...
'/usr/share/docker.io/contrib/nuke-graph-directory.sh' -> '/var/lib/docker/nuke-graph-dtrctory.sh'
Selecting previously unselected package aufs-tools.
(Reading database ... 212075 files and directories currently installed.)
Preparing to unpack .../aufs-tools_1%3a3.2+20130722-1.1ubuntu1_amd64.deb ...
Unpacking aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../docker-ce_17.06.1-ce-0~ubuntu_amd64.deb ...
Unpacking docker-ce (17.06.1-ce-0~ubuntu) ...
Setting up aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
Setting up docker-ce (17.06.1-ce-0~ubuntu) ...

```

9.3 Git

O GitHub, é um sistema de controle de versão distribuído. através do git, é possível trabalhar sempre em um mesmo diretório, muito utilizado em ambientes de produção, tendo em vista que é possível trabalhar em um mesmo projeto e se necessário retornar a versão anterior. Com isso, utilizaremos esse repositório para aplicação da nossa aplicação e automação de nosso ambiente. O GitHub, é um repositório publico, utilizado por desenvolvedores para hospedar seus projetos, no qual é utilizado o git para o controle de versões. Para instalação do git, utilizamos o comando a seguir.

\$ apt-get install git

Com o git instalado, podemos dar prosseguimento na implantação do nosso ambiente e temos o controle de versão.

10 IMPLANTAÇÃO DA APLICAÇÃO

Para a implantação da nossa aplicação, utilizaremos o Ansible para automação da instalação. Com isso, utilizaremos um playbook para instalação da estrutura da aplicação.

Para dar início, é necessário a criação de uma diretório aonde será implantada a aplicação, utilizando o comando;

```
$ mkdir ansibleodoo
```

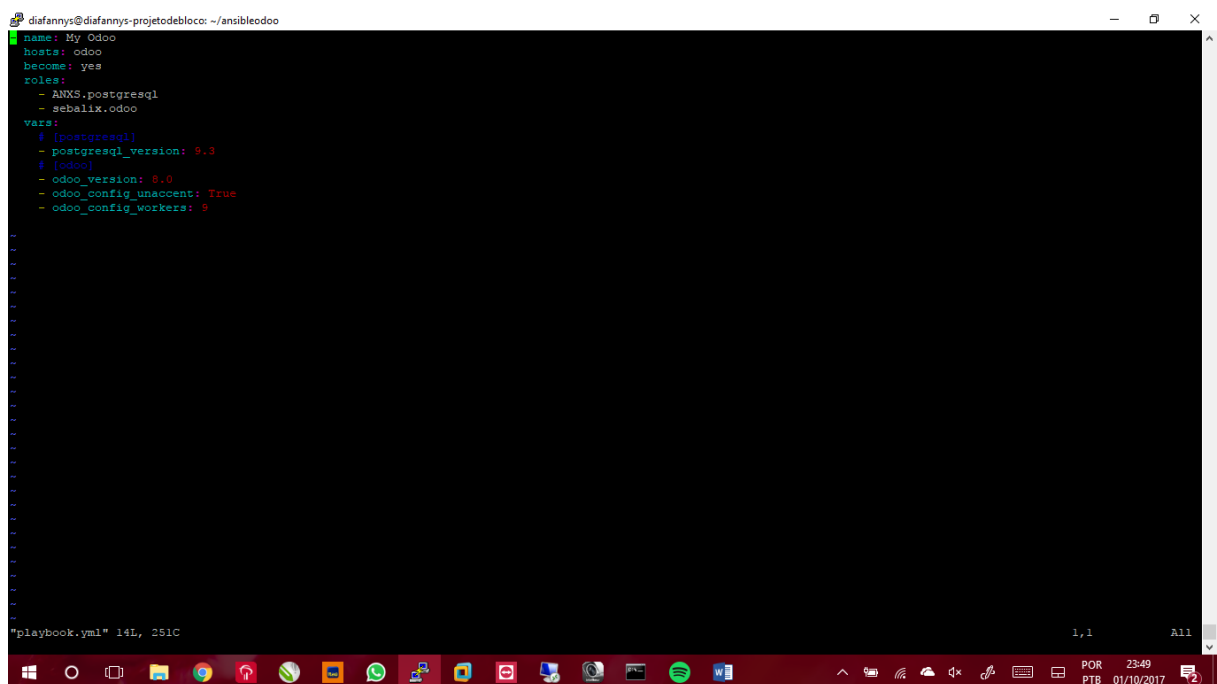
```
$ cd ansibleodoo
```

Com isso, entramos no diretório após a criação do mesmo. Para a execução do playbook, é necessário a criação de um arquivo YAML para execução desse serviço. Com os comandos a seguir, é criado um arquivo `playbook.yml` e em seguida aberto com o editor `vim` para adição das linhas de comando.

```
$ touch playbook.yml
```

```
$ vim playbook.yml
```

No playbook a seguir, podemos notar a nomeação do arquivo, os hosts de destino para execução do playbook, o apontamento para as roles de instalação do postgresql (banco de dados) e do odoo (aplicação), e com as variáveis a definição de versão que será instalada tanto do banco de dados, como da aplicação.



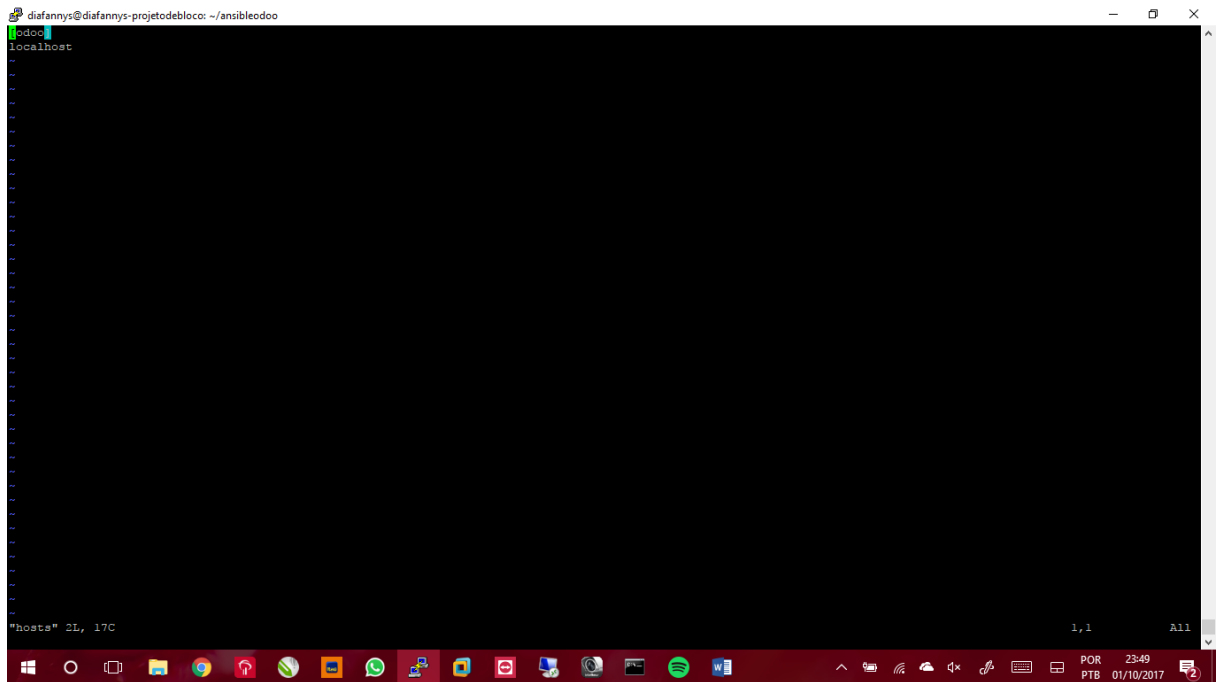
```
diafannys@diafannys-projetodebloco: ~/ansibleodoo
name: My Odoo
hosts: odoo
become: yes
roles:
  - ANXS.postgresql
  - sebalix.odoo
vars:
  postgresql:
    postgresql_version: 9.3
  odoo:
    odoo_version: 8.0
    odoo_config_unaccent: True
    odoo_config_workers: 9

"playbook.yml" 14L, 251C
1,1 All
```

Nessa etapa, é criado um arquivo nomeado hosts, aonde serão apontados os servidores aonde será implantado o ambiente da aplicação. Com os comandos em sequência.

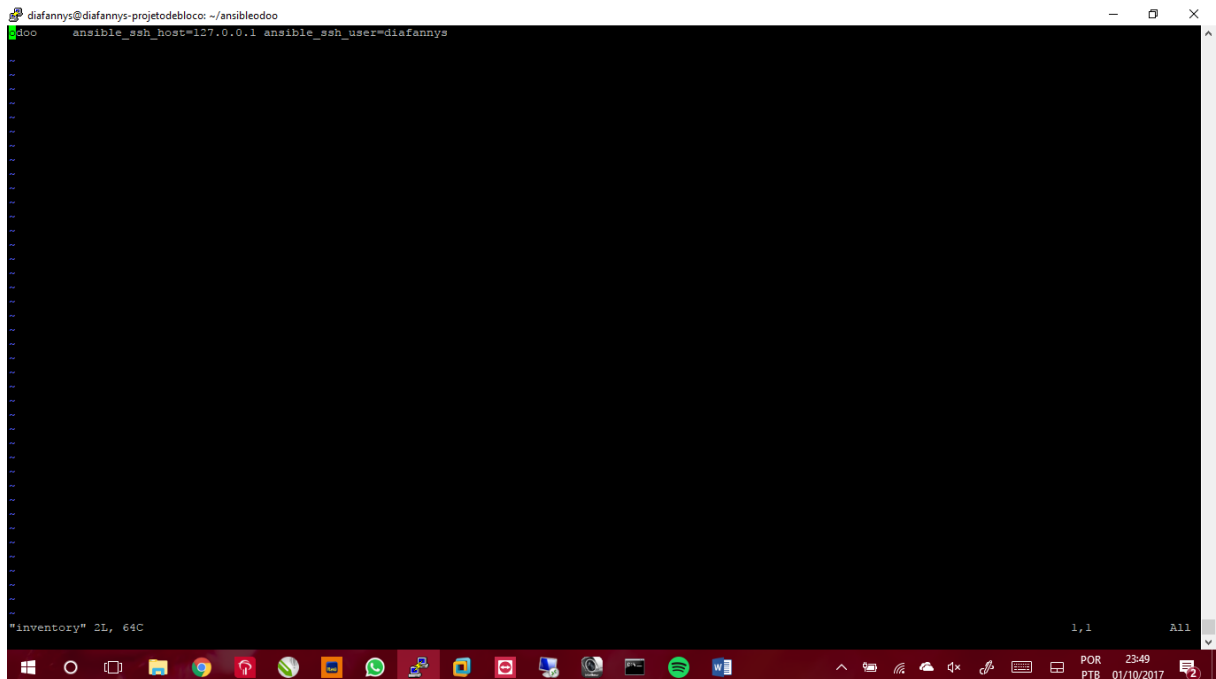
\$ touch hosts

\$ vim hosts



The screenshot shows a terminal window with a black background and green text. The prompt is `diafannys@diafannys-projetodebloco: ~/ansibleodoo`. The user has executed `touch hosts`, and the terminal shows `localhost` on the next line. The bottom status bar of the terminal indicates `"hosts" 2L, 17C` and `1,1 All`. The Windows taskbar is visible at the bottom with various application icons and a system clock showing 23:49 on 01/10/2017.

Após a criação do arquivo hosts, fazemos o mesmo processo com apontamento dentro do arquivo inventory.



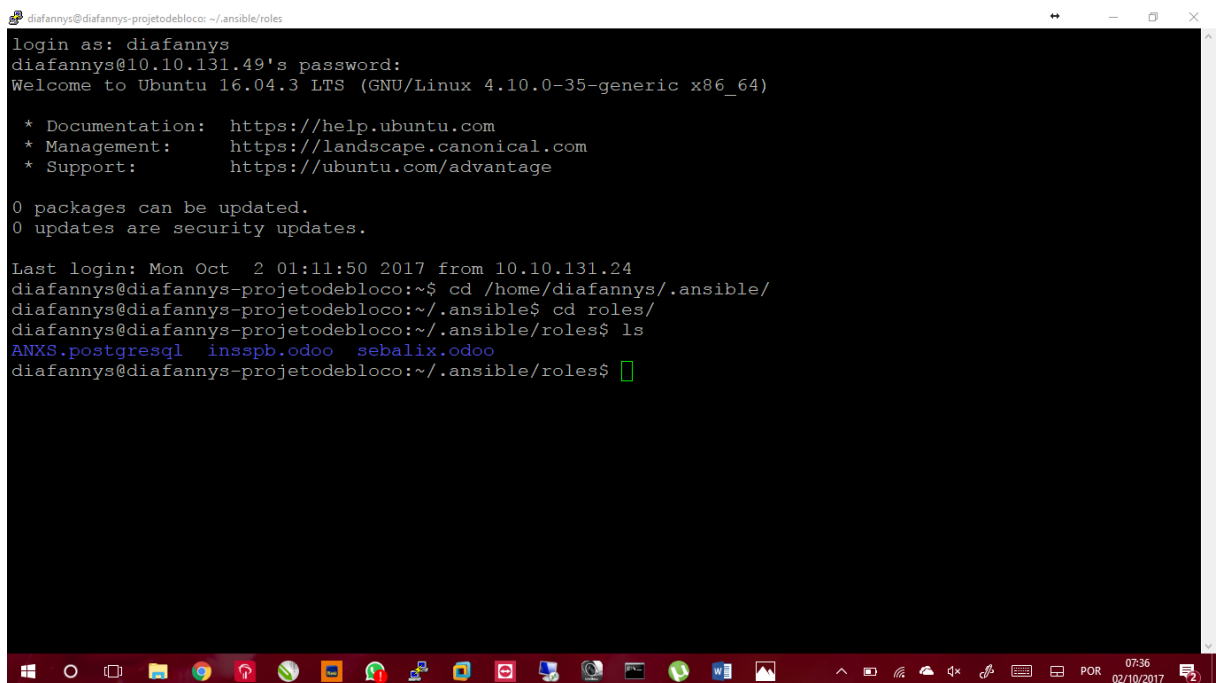
```

diafannys@diafannys-projetodebloco: ~/ansible/odoo
ansible_ssh_host=127.0.0.1 ansible_ssh_user=diafannys

"Inventory" 2L, 64C
1,1 All

```

E por fim, criadas as roles aonde serão apontadas as tarefas nos arquivos main.yml dentro de cada pasta destino das roles. Nesse projeto, as roles foram direcionadas para a pasta raiz do Ansible, com o caminho descrito na tela abaixo.



```

diafannys@diafannys-projetodebloco: ~/ansible/roles
login as: diafannys
diafannys@10.10.131.49's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-35-generic x86_64)

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

0 packages can be updated.
0 updates are security updates.

Last login: Mon Oct  2 01:11:50 2017 from 10.10.131.24
diafannys@diafannys-projetodebloco:~$ cd /home/diafannys/.ansible/
diafannys@diafannys-projetodebloco:~/.ansible$ cd roles/
diafannys@diafannys-projetodebloco:~/.ansible/roles$ ls
ANXS.postgresql  insspb.odoo  sebalix.odoo
diafannys@diafannys-projetodebloco:~/.ansible/roles$

```

Após o apontamento das roles escritos no Playbook, com o comando a seguir rodamos o playbook dentro da pasta com o playbook armazenado.

\$ Ansible-playbook -i hosts playbook.yml -e "ansible_ssh_user=diafannys" -k

```

diafannys@diafannys-projetodebloco: ~/ansibleodoo
TASK [ANXS.postgresql : PostgreSQL | Install PostgreSQL] *****
ok: [odoo] => (item=[u'postgresql-9.3', u'postgresql-client-9.3', u'postgresql-contrib-9.3'])

TASK [ANXS.postgresql : PostgreSQL | PGtune] *****
skipping: [odoo]

TASK [ANXS.postgresql : PostgreSQL | Make sure the CA certificates are available] *****
skipping: [odoo]

TASK [ANXS.postgresql : PostgreSQL | Add PostgreSQL repository] *****
skipping: [odoo]

TASK [ANXS.postgresql : PostgreSQL | Make sure the dependencies are installed] *****
skipping: [odoo] => (item=[])

TASK [ANXS.postgresql : PostgreSQL | Install PostgreSQL] *****
skipping: [odoo] => (item=[])

TASK [ANXS.postgresql : PostgreSQL | PGtune] *****
skipping: [odoo]

TASK [ANXS.postgresql : PostgreSQL | Extensions | Make sure the postgres contrib extensions are installed | Debian] *****
skipping: [odoo]

TASK [ANXS.postgresql : PostgreSQL | Extensions | Make sure the postgres contrib extensions are installed | RedHat] *****
skipping: [odoo]

TASK [ANXS.postgresql : PostgreSQL | Extensions | Make sure the development headers are installed | Debian] *****
skipping: [odoo]

TASK [ANXS.postgresql : PostgreSQL | Extensions | Make sure the development headers are installed | RedHat] *****
skipping: [odoo] => (item=[])

TASK [ANXS.postgresql : include vars] *****
skipping: [odoo] => (item=/home/diafannys/.ansible/roles/ANXS.postgresql/vars/../../vars/xenial.yml)

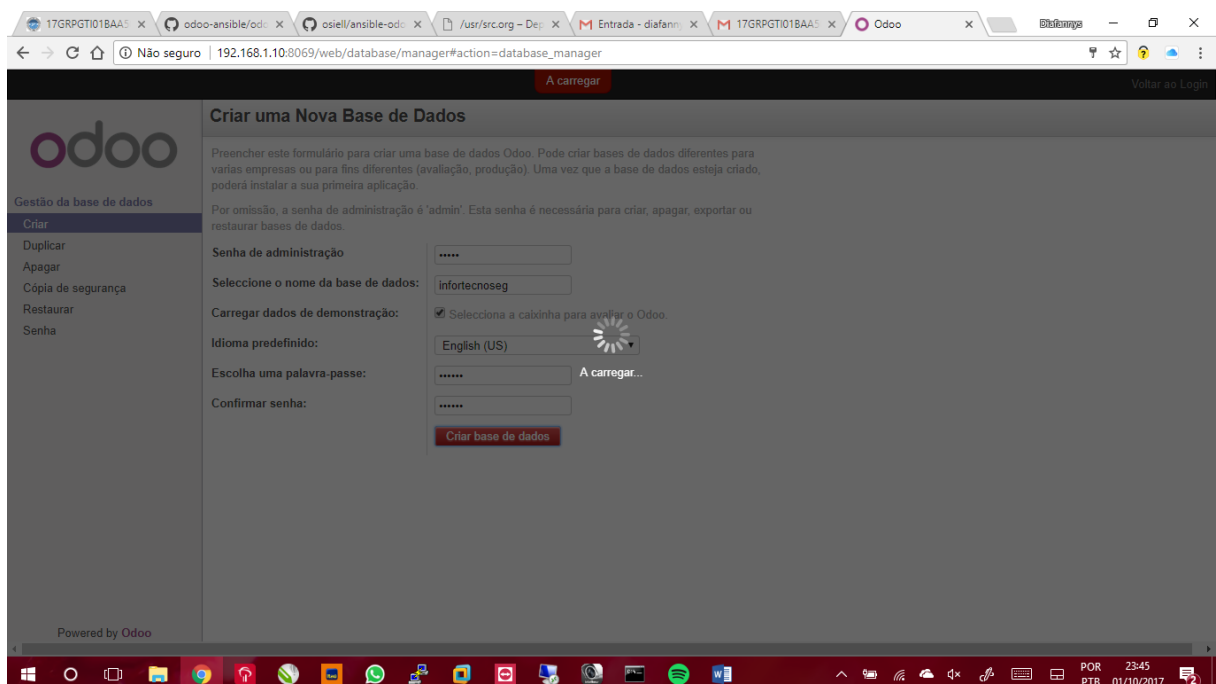
TASK [ANXS.postgresql : PostgreSQL | Extensions | Make sure the postgres extensions are installed] *****
skipping: [odoo] => (item=[])

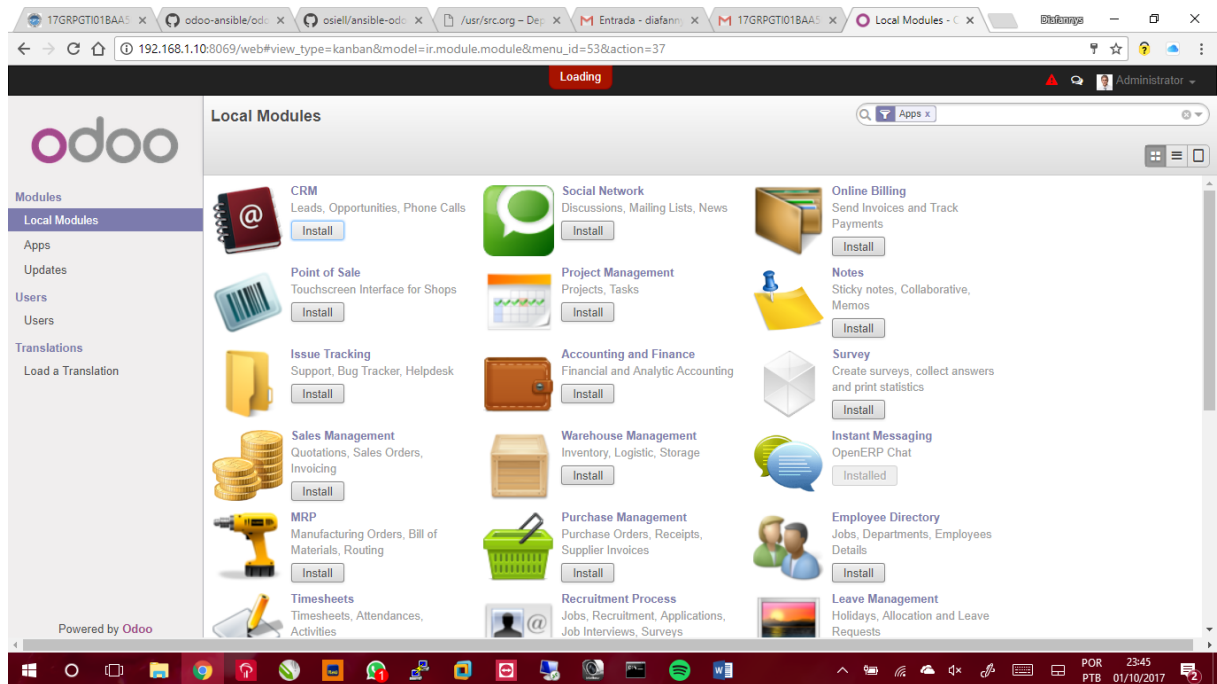
TASK [ANXS.postgresql : PostgreSQL | Drop the data directory | RedHat] *****
skipping: [odoo]

TASK [ANXS.postgresql : PostgreSQL | Make sure the postgres data directory exists] *****
ok: [odoo]

```

Após rodar o playbook, nossa aplicação está rodando, como demonstrado nas imagens a seguir. Na primeira imagem, é feito a configuração do baco de dados da aplicação, e em sequência, é feito o acesso da aplicação. Repositório disponível em <https://github.com/diafannys/projeto-odoo>



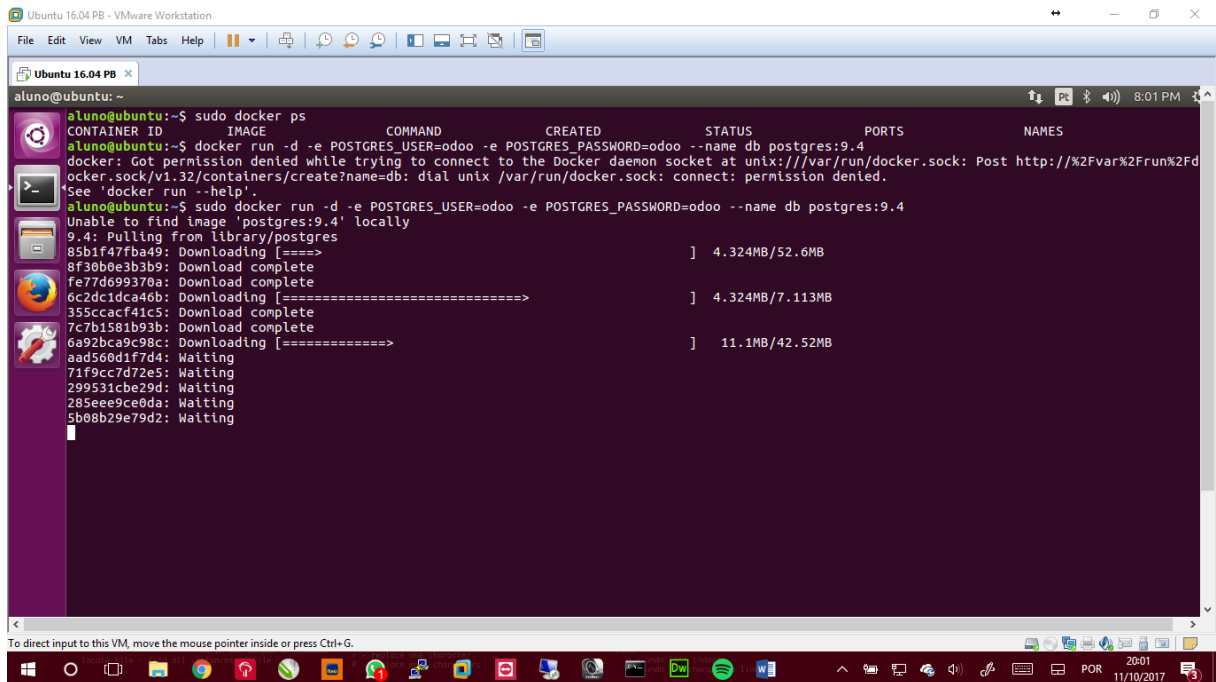


Após a instalação e configuração da aplicação. Vamos preparar a execução do Docker e com isso, podemos dar início a implantação dos containers do Odoo, que nesse ambiente foi selecionado a versão 10.0 e postgresql com a versão 9.4.

```
$ docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo -name db postgres:9.4
```

```
$ docker run -p 8069:8069 --name odoo --link db:db -t odoo
```

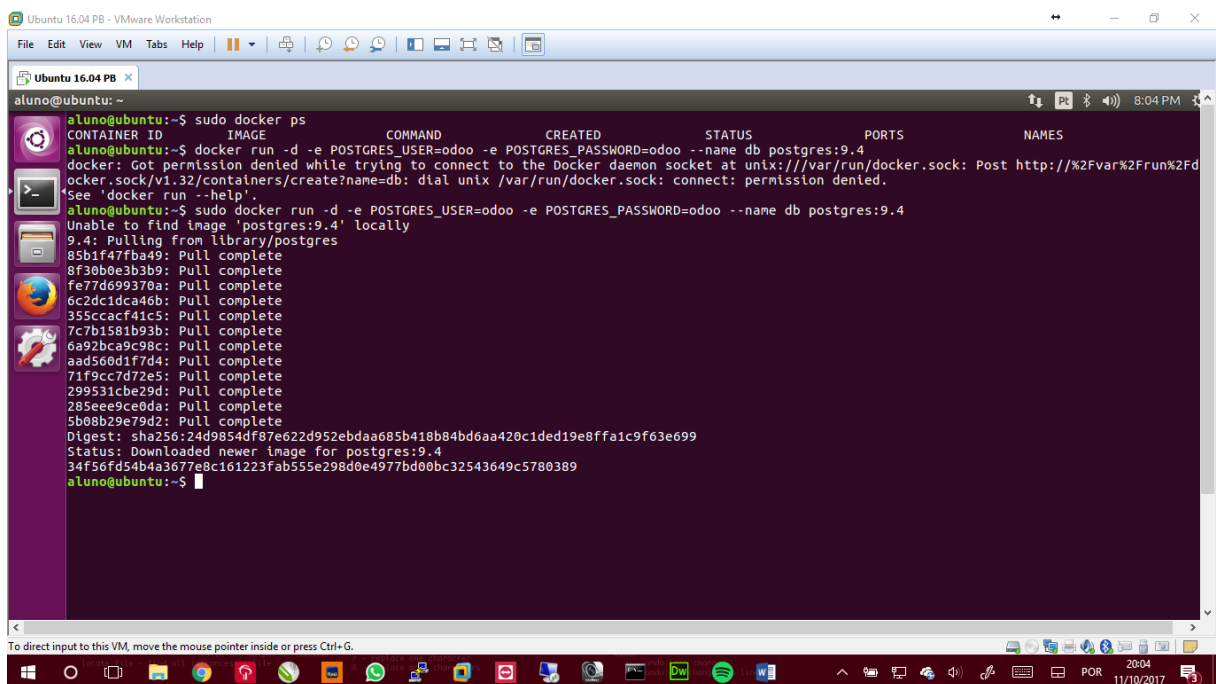
Abaixo, podemos verificar os containers em execução e por fim, a execução da aplicação, e acesso via navegador web.



```

aluno@ubuntu:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS              NAMES
aluno@ubuntu:~$ docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo --name db postgres:9.4
docker: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fd
ocker.sock/v1.32/containers/create?name=db: dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
aluno@ubuntu:~$ sudo docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo --name db postgres:9.4
Unable to find image 'postgres:9.4' locally
9.4: Pulling from library/postgres
85b1f47fba49: Downloading [====>] 4.324MB/52.6MB
8f30b0e3b3b9: Download complete
fe77d699370a: Download complete
6c2dc1dca46b: Downloading [=====] 4.324MB/7.113MB
355ccacf41c5: Download complete
7c7b1581b93b: Download complete
6a92bca9c98c: Downloading [=====] 11.1MB/42.52MB
aad560d1f7d4: Waiting
71f9cc7d72e5: Waiting
299531cbe29d: Waiting
285eee9ce0da: Waiting
5b08b29e79d2: Waiting

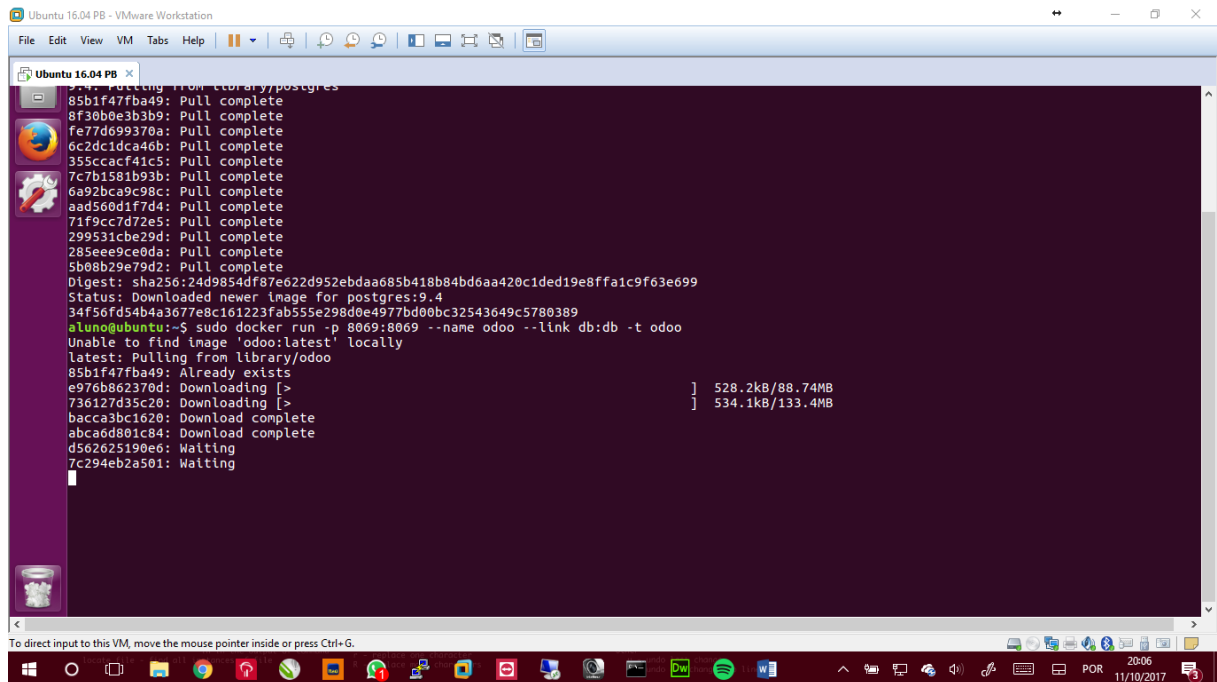
```



```

aluno@ubuntu:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS              NAMES
aluno@ubuntu:~$ docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo --name db postgres:9.4
docker: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fd
ocker.sock/v1.32/containers/create?name=db: dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
aluno@ubuntu:~$ sudo docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo --name db postgres:9.4
Unable to find image 'postgres:9.4' locally
9.4: Pulling from library/postgres
85b1f47fba49: Pull complete
8f30b0e3b3b9: Pull complete
fe77d699370a: Pull complete
6c2dc1dca46b: Pull complete
355ccacf41c5: Pull complete
7c7b1581b93b: Pull complete
6a92bca9c98c: Pull complete
aad560d1f7d4: Pull complete
71f9cc7d72e5: Pull complete
299531cbe29d: Pull complete
285eee9ce0da: Pull complete
5b08b29e79d2: Pull complete
Digest: sha256:24d9854df87e622d952ebd8a685b418b84bd6aa420c1ded19e8ffa1c9f63e699
Status: Downloaded newer image for postgres:9.4
34f56fd54b4a3677e8c161223fab555e298d0e4977bd00bc32543649c5780389
aluno@ubuntu:~$

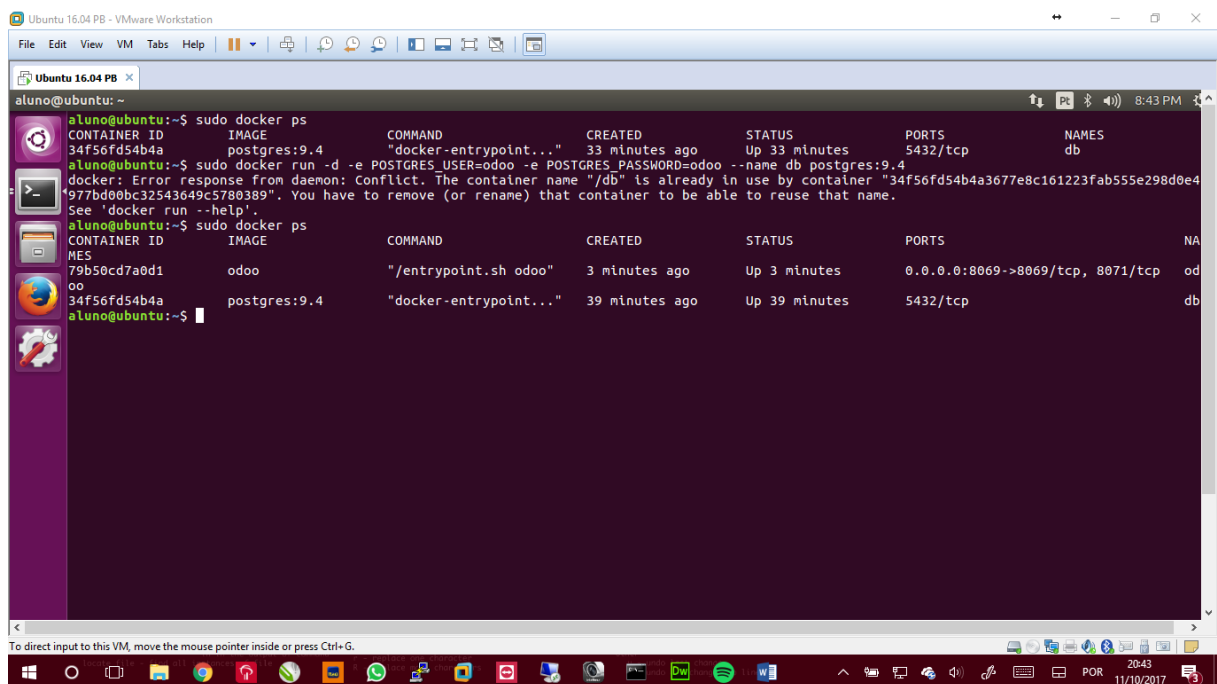
```



```

Ubuntu 16.04 PB - VMware Workstation
File Edit View VM Tabs Help
Ubuntu 16.04 PB x
85b1f47fba49: Pull complete
8f30b0e3b3b9: Pull complete
fe77d699370a: Pull complete
6c2dc1dca46b: Pull complete
355ccacf41c5: Pull complete
7c7b1581b93b: Pull complete
6a92bca9c98c: Pull complete
aad560d1f7d4: Pull complete
71f9cc7d72e5: Pull complete
299531cbe29d: Pull complete
285eee9ce0da: Pull complete
5b08b29e79d2: Pull complete
Digest: sha256:24d9854df87e622d952ebdaa685b418b84bd6aa420c1ded19e8ffa1c9f63e699
Status: Downloaded newer image for postgres:9.4
34f56fd54b4a3677e8c161223fab555e298d0e4977bd00bc32543649c5780389
aluno@ubuntu:~$ sudo docker run -p 8069:8069 --name odoo --link db:db -t odoo
Unable to find image 'odoo:latest' locally
latest: Pulling from library/odoo
85b1f47fba49: Already exists
e976b862370d: Downloading [>] 528.2kB/88.74MB
736127d35c20: Downloading [>] 534.1kB/133.4MB
bacca3bc1620: Download complete
abca6d801c84: Download complete
d562625190e6: Waiting
7c294eb2a501: Waiting

```



```

Ubuntu 16.04 PB - VMware Workstation
File Edit View VM Tabs Help
Ubuntu 16.04 PB x
aluno@ubuntu:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS               NAMES
34f56fd54b4a        postgres:9.4        "docker-entrypoint..." 33 minutes ago       Up 33 minutes       5432/tcp            db
aluno@ubuntu:~$ sudo docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo --name db postgres:9.4
docker: Error response from daemon: Conflict. The container name "/db" is already in use by container "/34f56fd54b4a3677e8c161223fab555e298d0e4977bd00bc32543649c5780389". You have to remove (or rename) that container to be able to reuse that name. See 'docker run --help'.
aluno@ubuntu:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS               NAMES
79b50cd7a0d1        odoo                "/entrypoint.sh odoo"    3 minutes ago        Up 3 minutes        0.0.0.0:8069->8069/tcp, 8071/tcp    odoo
34f56fd54b4a        postgres:9.4        "docker-entrypoint..." 39 minutes ago       Up 39 minutes       5432/tcp            db
aluno@ubuntu:~$

```

Ubuntu 16.04 PB - VMware Workstation

```

2017-10-12 01:36:42,967 1 INFO INFOFORTECNOSEG odoo.modules.registry: module web_kanban_gauge: creating or updating database tables
2017-10-12 01:36:43,929 1 INFO INFOFORTECNOSEG odoo.modules.loading: loading web_kanban_gauge/views/web_kanban_gauge_templates.xml
2017-10-12 01:36:43,976 1 INFO INFOFORTECNOSEG odoo.modules.registry: module web_settings_dashboard: creating or updating database tables
2017-10-12 01:36:43,105 1 INFO INFOFORTECNOSEG odoo.addons.auth_crypt.models.res_users: Hashing passwords, may be slow for databases with many u
sers....
2017-10-12 01:36:43,128 1 INFO INFOFORTECNOSEG odoo.modules.loading: loading web_settings_dashboard/views/dashboard_views.xml
2017-10-12 01:36:43,166 1 INFO INFOFORTECNOSEG odoo.modules.loading: loading web_settings_dashboard/views/dashboard_templates.xml
2017-10-12 01:36:43,279 1 INFO INFOFORTECNOSEG odoo.modules.loading: 12 modules loaded in 4.55s, 0 queries
2017-10-12 01:36:43,464 1 INFO INFOFORTECNOSEG odoo.modules.loading: Modules loaded.
2017-10-12 01:36:43,466 1 INFO INFOFORTECNOSEG odoo.models: Computing parent left and right for table ir_ui_menu...
2017-10-12 01:36:43,685 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:43] "POST /web/database/create HTTP/1.1" 303 -
2017-10-12 01:36:43,691 1 INFO INFOFORTECNOSEG odoo.addons.base.ir.ir_http: Generating routing map
2017-10-12 01:36:48,678 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:48] "GET /web/ HTTP/1.1" 200 -
2017-10-12 01:36:48,934 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:48] "GET /web/content/248-1a370e1/web.assets_backe
nd.1.css HTTP/1.1" 200 -
2017-10-12 01:36:48,979 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:48] "GET /web/content/249-05909b3/web_editor.summe
rnote.0.css HTTP/1.1" 200 -
2017-10-12 01:36:48,994 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:48] "GET /web/content/250-82e236e/web_editor.asset
s.editor.0.css HTTP/1.1" 200 -
2017-10-12 01:36:49,037 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:49] "GET /web/content/246-32fb078/web.assets_commo
n.0.css HTTP/1.1" 200 -
2017-10-12 01:36:49,099 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:49] "GET /web/content/247-1a370e1/web.assets_backe
nd.0.css HTTP/1.1" 200 -
2017-10-12 01:36:49,206 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:49] "GET /web/content/253-05909b3/web_editor.summe
rnote.js HTTP/1.1" 200 -
2017-10-12 01:36:49,272 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:49] "GET /web/content/254-82e236e/web_editor.asset
s.editor.js HTTP/1.1" 200 -
2017-10-12 01:36:49,403 1 INFO INFOFORTECNOSEG odoo.addons.base.ir.ir_attachment: filestore gc 253 checked, 0 removed
2017-10-12 01:36:49,414 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:49] "GET /web/content/251-32fb078/web.assets_commo
n.js HTTP/1.1" 200 -
2017-10-12 01:36:49,431 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:49] "GET /web/content/252-1a370e1/web.assets_backe
nd.js HTTP/1.1" 200 -
2017-10-12 01:36:49,441 1 INFO INFOFORTECNOSEG odoo.addons.base.ir.ir_autovacuum: GC'd 0 user log entries
2017-10-12 01:36:49,509 1 INFO INFOFORTECNOSEG werkzeug: 192.168.115.1 - - [12/Oct/2017 01:36:49] "GET /web/binary/company_logo HTTP/1.1" 200 -

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

CNA Manager - Alunos x Cursos: [17GRPQT01BAA501: Rol x library/odoo - Docker H- x Apps - Odoo x

192.168.115.145:8069/web/#view_type=kanban&model=ir.module.module&menu_id=52&action=37
















Apps Settings Administrator

odoo

Apps Updates

Apps Search...

1-35 / 35

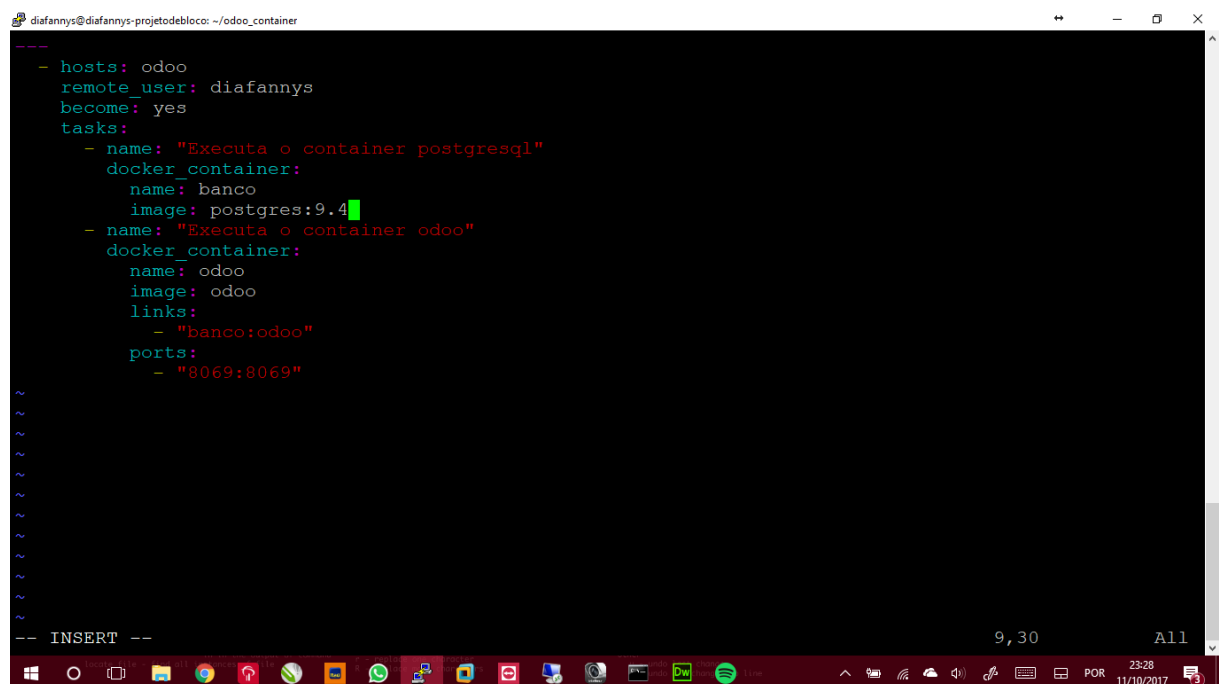
 CRM Leads, Opportunities, Activities Install	 Project Projects, Tasks Install	 Inventory Management Inventory, Logistics, Warehousing Install
 Manufacturing Manufacturing Orders, Bill of Materials, Routings Install	 Sales Quotations, Sales Orders, Invoicing Install	 Point of Sale Touchscreen Interface for Shops Install
 Timesheets Install	 Discuss Discussions, Mailing Lists, News Install	 Leave Management Holidays, Allocation and Leave Requests Install
 Invoicing Send Invoices and Track Payments Install	 Accounting and Finance Financial and Analytic Accounting Install	 Issue Tracking Support, Bug Tracker, Helpdesk Install
 Productivity Sticky notes, Collaborative, Memos Install	 Website Builder Build Your Enterprise Website Install	 eCommerce Sell Your Products Online Install

Powered by Odoo

Purchase Management Employee Directory Attendances

Como demonstrado na execução do playbook para instalação do odoo, nessa etapa, utilizaremos o Ansible para automatizar a instalação da aplicação distribuída odoo com o auxílio de um playbook escrito para instalação de dois containers, o postgresql e o odoo. Na descrição do playbook, podemos verificar a escolha da imagem do banco de dados, a configuração dos parâmetros para execução da aplicação, que vai desde a criação e execução dos containers, como o apontamento de usuário, banco de dados e senha, assim como o encaminhamento para a pasta destino aonde será provisionado o container da aplicação odoo, e por fim, a publicação do repositório no github.

https://github.com/diafannys/docker_odoo_infnet



```
diafannys@diafannys-projetodebloco: ~/odoo_container
--
- hosts: odoo
  remote_user: diafannys
  become: yes
  tasks:
    - name: "Executa o container postgresql"
      docker_container:
        name: banco
        image: postgres:9.4
    - name: "Executa o container odoo"
      docker_container:
        name: odoo
        image: odoo
        links:
          - "banco:odoo"
        ports:
          - "8069:8069"
-- INSERT --
```

The screenshot shows a terminal window with a dark background. The title bar indicates the user is 'diafannys' in a directory '~/odoo_container'. The terminal displays an Ansible playbook with two tasks: one to create a PostgreSQL container named 'banco' using the 'postgres:9.4' image, and another to create an Odoo container named 'odoo' using the 'odoo' image, linked to the 'banco' container and exposing port 8069. The bottom of the terminal shows a Windows taskbar with various application icons and a system clock indicating 9:30 on 11/10/2017.

```

diafannys@diafannys-projetodebloco: ~/odoo_container
TASK [Gathering Facts] *****
ok: [localhost]

TASK [Executa o container postgresql] *****
changed: [localhost]

TASK [Executa o container odoo] *****
ok: [localhost]

PLAY RECAP *****
localhost                : ok=3    changed=1    unreachable=0    failed=0

diafannys@diafannys-projetodebloco:~/odoo_container$ echo "# docker_odoo_infnet" >> README.md
diafannys@diafannys-projetodebloco:~/odoo_container$ git init
Initialized empty Git repository in /home/diafannys/odoo_container/.git/
diafannys@diafannys-projetodebloco:~/odoo_container$ git add README.md
diafannys@diafannys-projetodebloco:~/odoo_container$ git add *
diafannys@diafannys-projetodebloco:~/odoo_container$ git commit -m "primeiro commit, verificar acesso p
ostgre"
[master (root-commit) 3bc45df] primeiro commit, verificar acesso postgre
 4 files changed, 27 insertions(+)
 create mode 100644 README.md
 create mode 100644 hosts
 create mode 100644 odoocontainer.retry
 create mode 100644 odoocontainer.yml
diafannys@diafannys-projetodebloco:~/odoo_container$ git remote add origin https://github.com/diafannys
/docker_odoo_infnet.git
diafannys@diafannys-projetodebloco:~/odoo_container$ git push -u origin master
Username for 'https://github.com': diafannys
Password for 'https://diafannys@github.com':

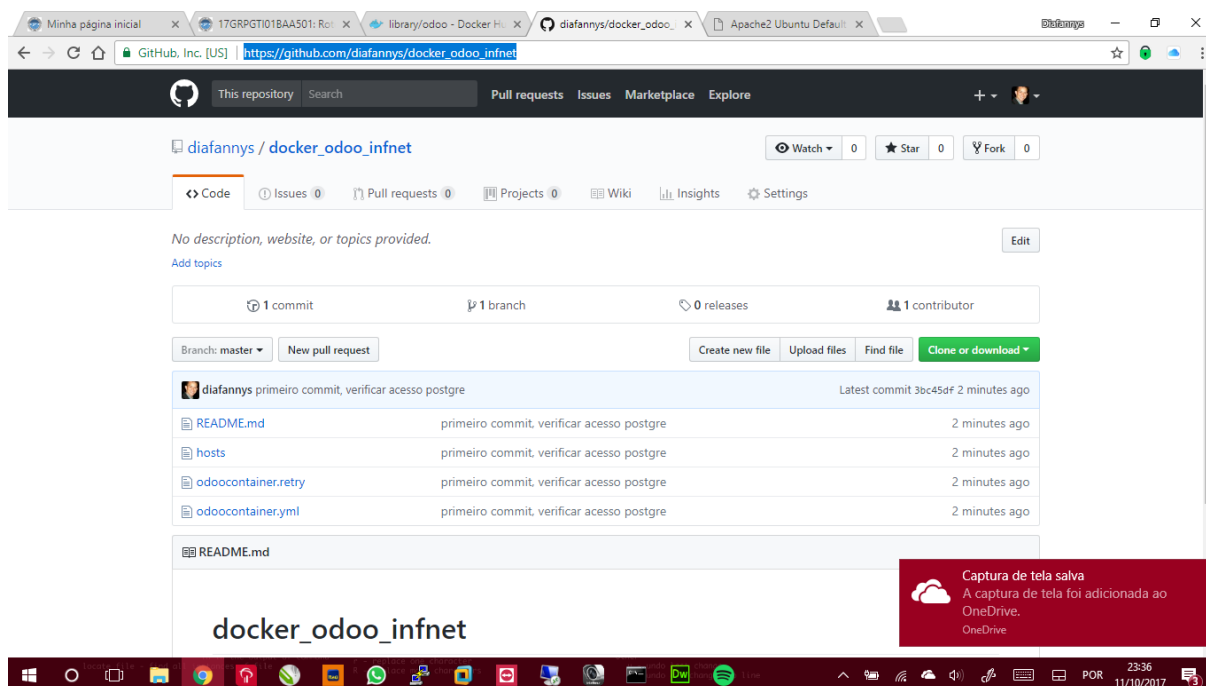
```

```

diafannys@diafannys-projetodebloco: ~/odoo_container
PLAY RECAP *****
localhost                : ok=3    changed=1    unreachable=0    failed=0

diafannys@diafannys-projetodebloco:~/odoo_container$ echo "# docker_odoo_infnet" >> README.md
diafannys@diafannys-projetodebloco:~/odoo_container$ git init
Initialized empty Git repository in /home/diafannys/odoo_container/.git/
diafannys@diafannys-projetodebloco:~/odoo_container$ git add README.md
diafannys@diafannys-projetodebloco:~/odoo_container$ git add *
diafannys@diafannys-projetodebloco:~/odoo_container$ git commit -m "primeiro commit, verificar acesso p
ostgre"
[master (root-commit) 3bc45df] primeiro commit, verificar acesso postgre
 4 files changed, 27 insertions(+)
 create mode 100644 README.md
 create mode 100644 hosts
 create mode 100644 odoocontainer.retry
 create mode 100644 odoocontainer.yml
diafannys@diafannys-projetodebloco:~/odoo_container$ git remote add origin https://github.com/diafannys
/docker_odoo_infnet.git
diafannys@diafannys-projetodebloco:~/odoo_container$ git push -u origin master
Username for 'https://github.com': diafannys
Password for 'https://diafannys@github.com':
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 657 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/diafannys/docker_odoo_infnet.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
diafannys@diafannys-projetodebloco:~/odoo_container$

```



11 CONCLUSÃO

Com o desenvolvimento deste projeto, podemos demonstrar a avaliação técnica no desenvolvimento do plano de implantação da aplicação, e o domínio referente as ferramentas adotadas na implantação do projeto. Foi apresentado pela diretoria de TI a necessidade de implantação de uma aplicação distribuída para atender a expansão do mercado da empresa Infortecnoseg, com isso, o departamento de Tecnologia da informação, propôs a implantação de um sistema distribuído, que será implantado através de máquinas virtuais em um ambiente de cloud Computing híbrido. Utilizando as ferramentas da Vmware, foi implantado um ambiente local para a implantação desses serviços, disponibilizando o acesso para a aplicação com maior qualidade e disponibilidade. Para complementar a distribuição da aplicação, foi escolhido a Amazon Web Services, aonde será expandido o acesso da aplicação com um ganho de disponibilidade do serviço e possível e necessária expansão futura do projeto. Com isso, podemos concluir que a implantação de uma nuvem híbrida, pode trazer um grande ganho para a infortecnoseg, por aumentar a disponibilidade dos serviços, e com isso o grau de segurança, tendo em vista as ferramentas disponíveis para expansão do serviço.

Com o intuito de ampliar os serviços e a disponibilidade da aplicação, podemos utilizar essa infraestrutura, e otimizar a implantação através de containers com o Docker, além de otimizar com a utilização do playbook escrito com Ansible.

12 BIBLIOGRAFIA

AWS VPC – Amazon Web Services -

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html

AWS VPC – Amazon Web Services -

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Scenario2.html

ESXi – VMware - <http://www.vmware.com> acessado em 05 de junho de 2017.

Odoo – Wikipédia - <https://pt.wikipedia.org/wiki/Odoo#Licen.C3.A7a> acessado em 17 de maio de 2017.

Installing Odoo – Odoo - <http://www.odoo.com/documentation/10.0/setup/install.html>

acessado em 17 de maio de 2017.

Windows Server 2016 containers – WRPD - <https://wrpd.com.br/windows-server-2016-containers-implantacao-e-configuracao/> acessado em 26 de junho de 2017.

Servidores Dell – DELL – <http://www.dell.com.br/servidores> acessado em 26 de Junho de 2017

<https://aws.amazon.com/pt/iam/>

<https://aws.amazon.com/pt/route53/>

<https://aws.amazon.com/pt/vpc/>

<https://aws.amazon.com/pt/ec2/>

<https://aws.amazon.com/pt/rds/>

<https://aws.amazon.com/pt/s3/>

<https://blog.octo.com/pt-br/virtualizacao-de-ambientes-provisionando-com-o-ansible-parte-3/>

<https://git-scm.com/book/id/v2/Getting-Started-Installing-Git>

<https://www.oficinadanet.com.br/post/14791-o-que-github>

<https://aws.amazon.com/pt/docker/>

<https://github.com/osiell/ansible-odoo>

<https://usr-src.org/blog/2015/03/deploy-odoo-with-ansible/>

https://github.com/ovnicraft/odoo-ansible/tree/master/odoo_ ansible/roles

<https://prezi.com/view/VYAICDV6RvvNbXKMdBxk/> (apresentação em sala)

<https://github.com/diafannys/>

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

MATERIAL DE ENSINO PRESENTES NO MOODLE E AULAS PRESENCIAS.