

GRADUAÇÃO EM REDE DE **COMPUTADORES**



Projeto de Bloco - Arquitetura e Infraestrutura de Aplicações [18E3_5] – Teste de Performance – TP09

Por: Gustavo de Alcantara Amaral

Aluno do Curso RDC, Turma 1, Manhã

Professor: Alexandre Carneiro

Professor orientador de Projeto de Bloco: Alexandre Carneiro

Tutor:

Rio de Janeiro
20/11/2018

Sumário

TP09 – Introdução:.....	3
Descrição	3
TP09 – Parte 1: Desenvolvimento	4
TP09 – Rubrica (Competência):.....	35
TP09 – Referências :	38

TP09 – Introdução:

Descrição

PARTE I - PRÁTICA

TP04

Documente a execução de uma aplicação distribuída, composta por dois ou mais containers Docker. Seu trabalho deve incluir um texto de pelo menos uma página explicando que aplicação vai ser executada e quais containers serão usados para isto. O restante da entrega deve incluir uma documentação completa da execução dos containers, e uma captura de tela mostrando o resultado alcançado.

A entrega deve ser feita no Moodle em um documento pdf.

TP06

Faça upload de uma versão inicial do capítulo de Introdução de seu Projeto de Bloco. Você deve contemplar:

O tipo de negócio/processo/problema que pretende tratar a partir de uma aplicação distribuída rodando sobre uma infraestrutura com virtualização.

Uma justificativa de porque este problema é relevante.

Uma descrição da aplicação distribuída que você pretende implementar com detalhes de sua arquitetura, como o gerenciamento do código-fonte, do processo de desenvolvimento, pré-requisitos para instalação, servidores necessários, etc.

Uma proposta inicial de como organizar a infraestrutura de sua aplicação, de acordo com o sistema de virtualização em que você pretende implementá-la.

A entrega deve ser feita no Moodle em um documento pdf.

TP07

Faça upload de uma versão inicial do capítulo 2/Proposta de Solução de seu Projeto de Bloco. Você deve contemplar:

Uma pequena descrição textual teórica sobre a arquitetura da solução de virtualização que você pretende utilizar.

Comparações entre a forma como você pretende implantar seu ambiente e outras abordagens, usando outras ferramentas.

Um planejamento passo a passo (com descrições de cada etapa) de como será feita a implantação da aplicação distribuída virtualizada.

Um cronograma estimado com o prazo para execução de cada atividade.

A entrega deve ser feita no Moodle em um documento pdf.

TP08

Faça upload de uma versão inicial do capítulo 3/Implementação de seu Projeto de Bloco.

Você deve contemplar:

Referências dos downloads e/ou versões de todos os componentes da solução implementada.
Todos os passos/capturas de tela de implementação de sua infraestrutura de virtualização.
Todos os passos/capturas de tela de configuração de sua aplicação.
Uma captura de tela de sua aplicação em funcionamento.
A entrega deve ser feita no Moodle em um documento pdf.

TP09

Faça upload do conteúdo completo de seu Projeto de Bloco até aqui, incluindo todos os capítulos apresentados nos TPs 6, 7 e 8, mais uma versão inicial do capítulo 4/Conclusões.

Além dos requisitos apresentados para os capítulos anteriores, suas conclusões devem incluir:

Se o prazo estabelecido no início do projeto foi adequado para a execução (não há problema se levou mais tempo que o planejado, mas se levou, explique as razões).

Se os recursos planejados (quantidade de memória, disco, etc.) foram suficientes para colocar a solução em funcionamento. Se não foram, explique as razões e sugira uma nova configuração.

Se as funcionalidades previstas na solução original funcionaram a contento. Se não foram, explique as razões.

Quais seriam as melhorias futuras que poderiam ser feitas no projeto executado, adotando soluções adicionais, aperfeiçoando instalações ou tornando o cenário mais complexo.

Antes da entrega final, todo o conteúdo de seu trabalho (mesmo que sejam apenas documentos do Word e PDFs) deve ser armazenado em um repositório GitHub, cujo link deve estar contido no corpo do trabalho.

A entrega deve ser feita no Moodle em um documento pdf.

TP09 – Parte 1: Desenvolvimento

PARTE I - PRÁTICA

Introdução:

Este documento tem o propósito de demonstrar a aplicabilidade de soluções distribuídas que serão implementadas e utilizadas futuramente em soluções de virtualização.

Objetivos:

Este documento tem como principal objetivo descrever a implementação da solução distribuída Wordpress em sistemas de nuvem, para atendimento das demandas da empresa XPTO.

A implementação levará em conta a tecnologia de virtualização baseada em container, automatização da implementação através de uma solução compatível (Ansible).

O objetivo final é que após a implementação esteja totalmente funcional em seu ambiente local, a mesma seja implementada em uma nuvem pública (Amazon AWS) utilizando a funcionalidade de Imagens de máquina da Amazon (AMIs).

Desenvolvimento:

- 1- Descrição do problemas e apresentação de uma aplicação distribuída para implementação na Empresa XPTO. Descrição da empresa onde será implementada a solução distribuída, detalhamento dos processos/atividades/pessoas que serão atendidas pela implementação e que tipo de benefícios se espera produzir com este projeto:**

A empresa XPTO tem um site que é implementado e atualizado com o Wordpress.

Para suportar este serviço a área de TI da empresa XPTO possui:

- a- Ambiente de produção com servidores em Alta disponibilidade;
- b- Ambiente de testes (laboratório);
- c- Sistema de backups em storage;
- d- Firewall

Toda a estrutura existe fisicamente e é suportada pelo time local de TI.

Após algumas falhas críticas e perdas de faturamento devido ao tempo fora de serviço e de restabelecimento, ocorreu uma decisão gerencial para implementação de uma solução virtualizada em nuvem (pública ou privada – A ser decidido).

A estrutura lógica existente:

Sistema Operacional – Linux Ubuntu Server 16.5 LTS;
Aplicação distribuída – Wordpress
Banco de dados – MySQL 5
Linguagem – PHP Versão 7
Servidor http: Apache

A estrutura física existente:

O cenário atual é implementado em uma estrutura de servidores físicos localizados no datacenter da empresa XPTO.

A empresa possui boas práticas de gerenciamento de TI e gestão de processos, mas, falhas ocorreram que causaram grande prejuízo financeiros forçando a tomada de decisão no sentido da implementação de uma solução virtualizada.

Requisitos da implementação:

- Implementar Wordpress (solução distribuída);
- Deve-se implementar a solução em uma instancia virtualizada utilizando uma nuvem pública ou privada;
- Deve-se utilizar a tecnologia de container para facilitar a recuperação do sistema operacional, banco de dados e da aplicação;
- A solução deve suportar o sistema operacional Ubuntu Server 16.5 LTS (Long Term Support);

Apresentação da aplicação solução distribuída virtualizada que será implantada:

Para suportar a aplicação distribuída escolhida será implementada uma estrutura da seguinte forma:

Nuvem pública:

O que é a computação em nuvem:

A computação em nuvem é a entrega sob demanda de poder computacional, armazenamento de banco de dados, aplicações e outros recursos de TI por meio de uma plataforma de serviços de nuvem via Internet com uma definição de preço conforme o uso.

Fundamentos da computação em nuvem

Trata-se de uma plataforma de serviços em nuvem que oferece acesso rápido a recursos de TI flexíveis e de baixo custo. Com a computação em nuvem, não é preciso realizar grandes investimentos iniciais em hardware e perder tempo nas atividades de manutenção e gerenciamento desse hardware. Ao invés disso, é possível provisionar exatamente o tipo e tamanho corretos de recursos computacionais necessários para executar a demanda solicitada pelo departamento de TI.

Como a computação em nuvem funciona:

A computação em nuvem oferece uma forma simples de acessar servidores, armazenamento, bancos de dados e um conjunto amplo de serviços de aplicativos via Internet. Uma plataforma de serviços em nuvem, é proprietária e faz a manutenção do hardware conectado à rede necessário para esses serviços de aplicativos, enquanto você provisiona e utiliza o que precisa por meio de um aplicativo web.

Vantagens e benefícios da computação em nuvem:

Substitua despesas de capital por despesas variáveis:

Ao invés de investir substancialmente em datacenters e servidores antes de saber como serão utilizados, a computação em nuvem permite consumir recursos de computação, e pagar apenas pela quantidade consumida.

Beneficie-se de economias massivas de escala:

Ao utilizar a computação em nuvem, pode-se alcançar um custo variável mais baixo do que seria possível normalmente. Como a utilização de centenas de milhares de clientes é agregada na nuvem, provedores de serviços em nuvem conseguem alcançar economias de escala maiores, o que se traduz em preços mais baixos com pagamento conforme o uso.

Elimine a ociosidade da capacidade implementada:

Elimine as suposições ao determinar a necessidade de capacidade de infraestrutura. Ao tomar uma decisão sobre a capacidade, antes da implementação do aplicativo, é frequentemente lidar com a ociosidade de recursos caros ou com limites de capacidade. Com a computação em nuvem, esses problemas desaparecem. A empresa pode acessar o quanto precisar, e escalonar para maior ou para menor conforme necessário com apenas alguns minutos de antecedência.

Aumente a velocidade e agilidade:

No ambiente de computação em nuvem, recursos adicionais de TI estão ao alcance em apenas um clique, o que significa que o tempo necessário para disponibilizar estes recursos aos desenvolvedores é reduzido de semanas para apenas minutos. Isso resulta em um aumento dramático na agilidade da organização, pois o custo e tempo necessários para experimentar e desenvolver é significativamente mais baixo.

A computação em nuvem permite eliminar o gasto de dinheiro com execução e manutenção de datacenters.

Com isto, as empresas podem se concentrar em projetos que diferenciam estas empresas, ao invés de focar na infraestrutura. A computação em nuvem permite que a empresa se volte aos seus clientes, ao invés do trabalho pesado de estruturar, empilhar e manter servidores ligados.

Acesso global em minutos:

Implante facilmente o aplicativo de uma empresa em várias regiões em todo o mundo com apenas alguns cliques. Isso significa que é possível oferecer latência menor e uma experiência melhor aos clientes de forma simples e por um custo mínimo.

Tipos de computação em nuvem:

A computação em nuvem tem três tipos principais que são comumente chamados de Infraestrutura como Serviço (IaaS), Plataforma como Serviço (PaaS) e Software como Serviço (SaaS). A seleção do tipo certo de computação em nuvem para as necessidades de uma empresa, pode ajudá-la a encontrar o equilíbrio certo de controle e evitar trabalho pesado indiferenciado.

Modelos de computação em nuvem:**Infraestrutura como um serviço (IaaS):**

A infraestrutura como um serviço, abreviada como IaaS, contém os componentes básicos da TI em nuvem e, geralmente, dá acesso (virtual ou no hardware dedicado) a recursos de rede e computadores, como também espaço para o armazenamento de dados. A infraestrutura como um serviço oferece a você o mais alto nível de flexibilidade e controle de gerenciamento sobre os seus recursos de TI e se assemelha bastante aos recursos de TI atuais com os quais muitos departamentos de TI e desenvolvedores estão familiarizados hoje em dia.

Plataforma como um serviço (PaaS):

Com as plataformas como um serviço, as empresas não precisam mais gerenciar a infraestrutura subjacente (geralmente, hardware e sistemas operacionais), permitindo que você se concentre na implantação e no gerenciamento das suas aplicações. Isso o ajuda a tornar-se mais eficiente, pois elimina as suas preocupações com aquisição de recursos, planejamento de capacidade, manutenção de software, patching ou qualquer outro tipo de trabalho pesado semelhante envolvido na execução da sua aplicação.

Software como um serviço (SaaS):

O software como um serviço oferece um produto completo, executado e gerenciado pelo provedor de serviços. Na maioria dos casos, as pessoas que se referem ao software como um serviço estão se referindo às aplicações de usuário final. Com uma oferta de SaaS, não é necessário pensar sobre como o serviço é mantido ou como a infraestrutura subjacente é gerenciada, você só precisa pensar em como usará este tipo específico de software. Um exemplo comum de aplicação do SaaS é o webmail, no qual você pode enviar e receber e-mails sem precisar gerenciar recursos adicionais para o produto de e-mail ou manter os servidores e sistemas operacionais no qual o programa de e-mail está sendo executado.

Para implementação de nuvem pública a escolha recomendada seria a utilização do servidor Amazon AWS;

Para implementação de nuvem privada a escolha recomendada seria a utilização da solução Openstack;

Para atendimento as demandas da empresa XPTO, a solução escolhida foi a solução em nuvem Pública oferecida pela Amazon AWS. Os principais motivos para esta escolha são:

- **Amazon é líder no mercado de provedores de nuvem Pública;**
- **Segurança – Amazon AWS possui diversas certificações, muitas voltadas para a áreas de segurança de dados, tais como:**
 - SOC 1/ISAE 3402;
 - SOC 2;
 - SOC 3;
 - Nível 1 do PCI DSS;
 - ISO 27001;
 - IRAP;
 - FIPS 140-2;
 - MPAA;
 - HIPAA;
 - FedRAMP (SM);
 - DoD CSM níveis 1-2, 3-5;
 - DIACAP e FISMA;
 - Certificação MTCS nível 3;
 - ITAR;
 - CSA;
 - ISO 9001;
- **Gerenciamento de identidade e acesso – Amazon diversas tecnologias para garantir o acesso seguro as instâncias criadas, tais como:**
 - Gerenciamento de usuários e grupos;
 - Integração com sistemas de gerenciamento de identidade e acesso existentes;
 - Gerenciamento de chaves dedicado e baseado em hardware
- **Infraestrutura global – A Amazon AWS dispõe de uma infraestrutura geograficamente espalhada em diversos locais do mundo, produzindo benefícios, tais como:**
 - Entrega de conteúdo com baixa latência;
 - Resolução de nomes de domínio confiável e com baixa latência;
- **Custos – A Amazon AWS dispõe de uma serie de facilidades envolvendo o gerenciamento de custos dos seus projetos de virtualização em nuvem Pública, tais como:**
 - Abordagem de definição de preço que oferece custos menores à medida que a demanda por mais capacidade na nuvem cresce, oferecendo custos ainda mais baixos com a reserva de capacidade;

- Escalabilidade – Provisionamento da capacidade de computação necessária da forma e no momento que é necessário;
- Opções abrangentes de configuração de rede, tais como:
 - Atribua facilmente endereços IP públicos estáticos às instâncias;
 - Controle de configuração de redes IP;
 - Criação de várias sub-redes privadas e públicas;
 - Anexar várias interfaces de rede a recursos de computação;
 - Controle o tráfego de entrada e saída dos recursos de computação de forma granular;
 - Conexão de redes privadas virtuais baseadas em hardware aos recursos da nuvem;
 - Alta velocidade, baixa latência, conectividade privada e dedicada entre infraestruturas no local e na nuvem;

2- Apresentar a organização da infraestrutura que será implementada, com diagrama de rede contendo os elementos de virtualização, detalhes sobre a rede:

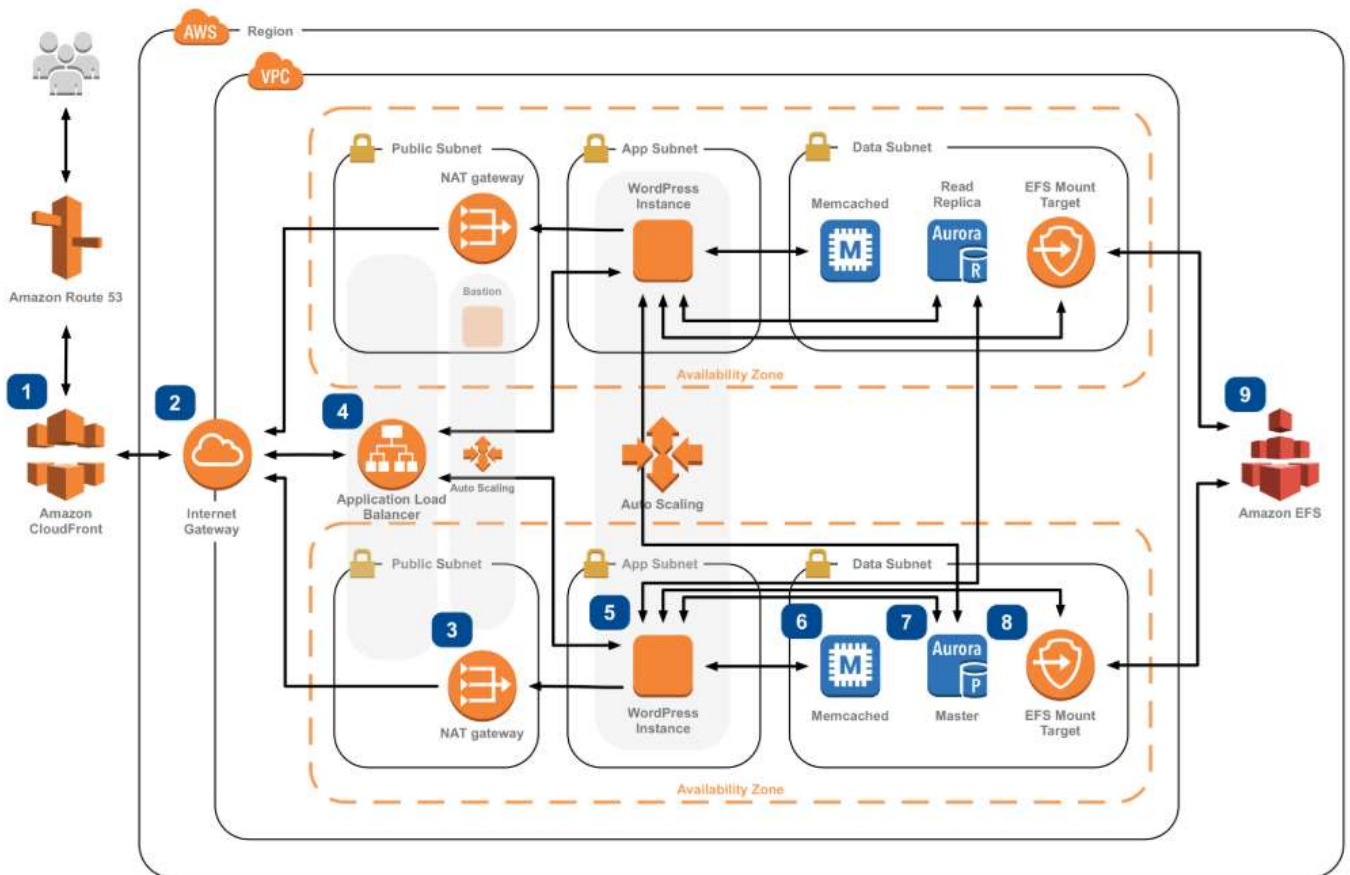


Diagrama acima para uma implementação utilizando o Amazon AWS como uma nuvem pública e infraestrutura de um provedor na nuvem;

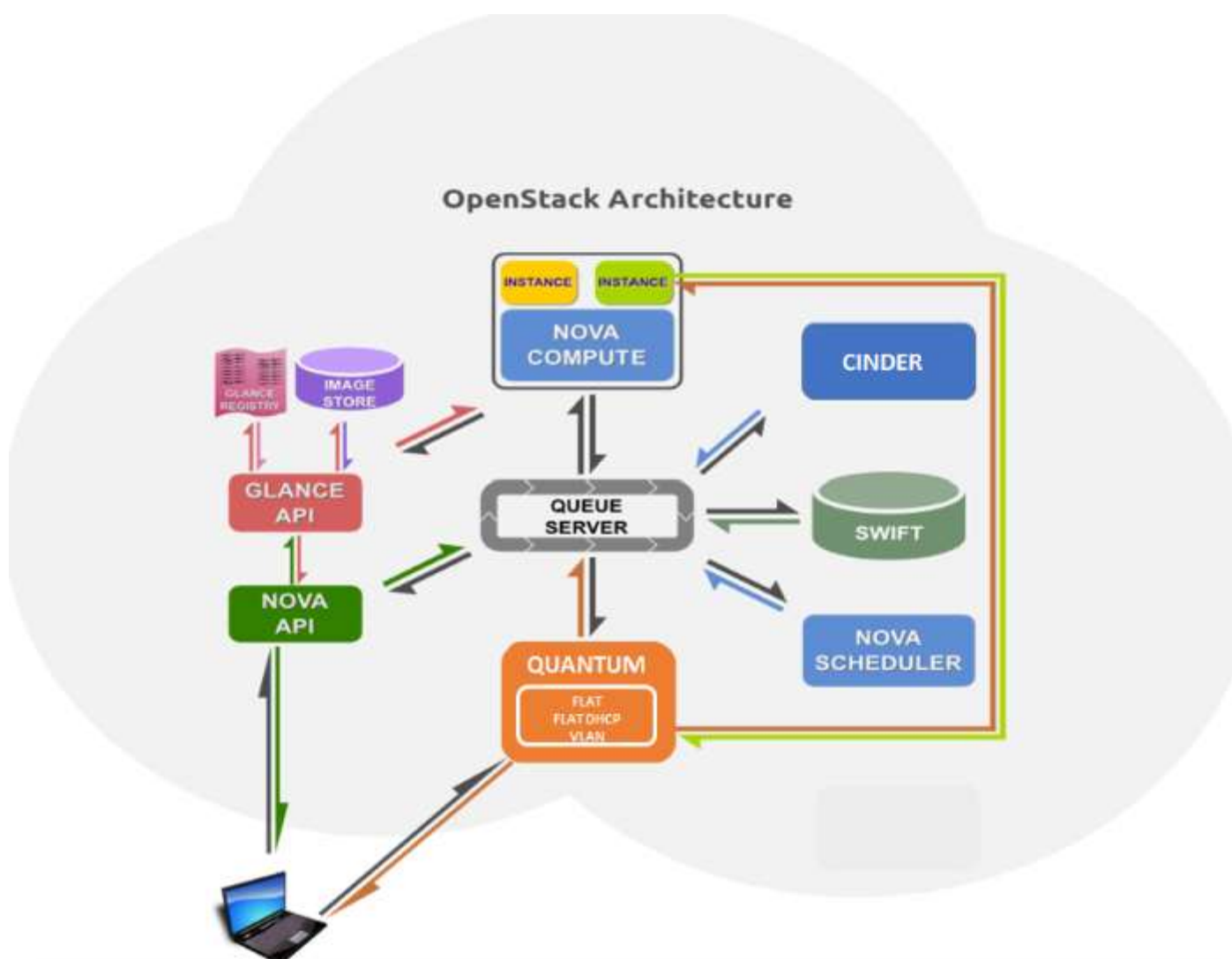
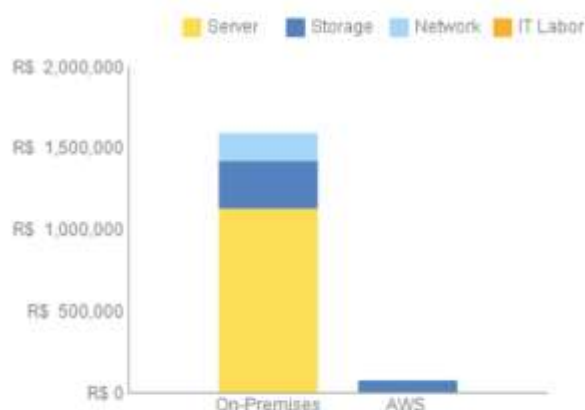


Diagrama acima para uma implementação utilizando o OpenStack dentro de uma nuvem privada com infraestrutura própria;

- 3- Descrever as principais referências sobre as tecnologias abordadas nesta implementação, informando o tipo de virtualização, edições, recursos e custos aproximados:

Custos aproximados de uma implementação em nuvem pública:

3 Years Cost Breakdown



3 Yr. Total Cost of Ownership		
	On-Premises	AWS
Server	R\$ 1,135,115	R\$ -
Storage	R\$ 291,127	R\$ 72,022
Network	R\$ 163,693	R\$ -
IT-Labor	R\$ -	R\$ -
Total	R\$ 1,589,935	R\$ 72,022

AWS cost includes business level support

Your On-Premises environment

Environment : Virtual					
# of VMs	vCPU	RAM (GB)	OS	Avg. Utilization	Optimize by
10	8	32	Linux	100%	RAM
10	8	32	Linux	100%	RAM

Storage (TB)		
SAN	NAS	Object
10	0	0

Fonte: <https://awstccalculator.com/#>

Custos aproximados de uma implementação em nuvem privada:

A priori não haveriam custos, pois, seria utilizada a infraestrutura existente.

4- Realizar a comparação da solução de virtualização utilizada no projeto e as demais:

Nuvem:

Conforme já explicado, uma aplicação baseada na nuvem é totalmente implantada na nuvem e todos os aspectos da aplicação são executados nela. As aplicações na nuvem ou foram criadas nela ou foram migradas de uma infraestrutura prévia para usufruírem dos benefícios da computação em nuvem. As aplicações baseadas na nuvem podem se beneficiar de fragmentos secundários da infraestrutura ou podem utilizar serviços de nível superior que reduzem as necessidades de gerenciamento, arquitetura e escalabilidade da infraestrutura principal.

Híbrida:

Uma implantação híbrida é uma maneira de conectar infraestrutura e aplicações entre recursos da Web e recursos atuais que não se encontram na nuvem. O método mais comum de implantação híbrida é o que ocorre entre a nuvem e a infraestrutura local atual para estender e aumentar a infraestrutura de uma empresa na nuvem enquanto recursos da nuvem são conectados ao sistema interno.

Local:

A implantação de recursos locais, usando ferramentas de gerenciamento de virtualização e recursos, às vezes é chamada de "nuvem privada". A implantação local não oferece muitos dos benefícios da computação em nuvem, mas, às vezes, é preferida por sua capacidade de oferecer recursos dedicados. Na maioria dos casos, este modelo de implantação é igual à infraestrutura de TI antiga, pois usa tecnologias de gerenciamento e virtualização de aplicações para tentar aumentar a utilização de recursos.

5- Apresentar o planejamento da instalação, com os passos a serem realizados, na ordem em que devem acontecer:

Etapa 1: Executar uma instância do Amazon EC2:

Etapa 2: Configurar uma instância:

Etapa 3: Alterações no site:

Etapa 4: Associe um nome de domínio ao site usando o Amazon Route 53:

Etapa 5: Obtenha ou migre um URL estático:

Etapa 6: Registre ou migre um nome de domínio:

Etapa 7: Configuração de DNS:

Etapa 8: Associar o Amazon EC2:

6- Apresentar o cronograma para implantação da solução pretendida:

WBS	SEMANA 01	SEMANA 02	SEMANA 03	SEMANA 04	SEMANA 05	SEMANA 06	SEMANA 07	SEMANA 08	SEMANA 09	SEMANA 10	SEMANA 11	SEMANA 12
Realizar o Planejamento:												
Realizar o Kick off do Projeto:												
Realizar a Implantação da solução virtualizada:												
Realizar os testes operacional:												
Realizar o handover do ambiente de testes para produção:												
Realizar o aceite formal da solução virtualizada:												
Realizar os treinamentos das equipes de Manutenção e Operação:												
Realizar os treinamentos dos usuários da solução:												

7- Detalhamento da solução implementada:

a. Sistema Operacional:



Ubuntu é um sistema operacional Linux, open source, baseado no Debian e desenvolvido pela empresa Canonical Ltd.

O Ubuntu 16.4 LTS será o sistema operacional que será utilizado como plataforma para executarmos todos os processos inerente a solução Wordpress.

b. MySQL:

MySQL é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. O serviço utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

O MySQL será utilizado como banco de dados para o sistema Wordpress que será utilizado neste projeto.

c. Apache:

Criado em 1995 por Rob McCool, na época funcionário da NCSA (National Center for Supercomputing Applications), o servidor Apache ou Servidor HTTP Apache é o mais bem sucedido servidor web livre que existe. Trata-se de um servidor web muito popular, utilizado principalmente no Linux.

Será utilizado o Servidor Apache para prover o blog da empresa XPTO na internet.

d. Docker:

Docker é uma plataforma Open Source escrito em Go, que é uma linguagem de programação de alto desempenho desenvolvida dentro do Google, que facilita a criação e administração de ambientes isolados

Isso reduz drasticamente o tempo de deploy de alguma infraestrutura ou até mesmo aplicação, pois não há necessidade de ajustes de ambiente para o correto funcionamento do serviço, o ambiente é sempre o mesmo, configure-o uma vez e replique-o quantas vezes quiser.

Outra facilidade do Docker é poder criar suas imagens (containers prontos para deploy) a partir de arquivos de definição chamados Dockerfiles (veremos isso melhor em posts futuros).

A tecnologia de container será demonstrada neste projeto.

e. Ansible:

Ansible é uma plataforma de automação simples de usar, robusta e totalmente de código aberto. Ansible está disponível gratuitamente e é executado em Linux, Mac ou BSD.

O Ansible será utilizado neste projeto para automação das implementações.

f. GIT & Github:

GIT – Git é um dos vários Sistemas de Controle de Versão de arquivos. Através dele é possível desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas. Uma das principais aplicações do Git é justamente esta, permitir que um arquivo possa ser editado ao mesmo tempo por pessoas diferentes.

Github - GitHub é uma espécie de rede social para desenvolvedores e programadores. Ele possui diversas funcionalidades, como feeds, followers (que seriam seus amigos), wiki e um gráfico que mostra como os desenvolvedores estão contribuindo trabalhando nas versões de seus repositórios. Este serviço web oferece diversas funcionalidades extras aplicadas ao Git, de forma gratuita, mas também existem planos comerciais para projetos de código privados. O GitHub permite que você hospede seus projetos pessoais em nuvem. Além disso, quase todos os projetos/frameworks/bibliotecas sobre desenvolvimento open source estão no GitHub, e você pode acompanhá-los através de novas versões, contribuir informando bugs ou até mesmo enviando código e correções.

Neste projeto, será utilizado os comando Gits e o repositório do Github para compartilhamento e contribuição de equipes para este projeto.

g. Wordpress:

Trata-se de uma solução distribuída para criação de sites “www” que pode ser utilizada desde a criação de blogs na internet, até o desenvolvimento de sites para e-commerces.

Atualmente, cerca de 25% de toda internet utiliza o Wordpress para algum serviço.

O Wordpress será utilizado neste projeto como solução distribuída a ser implementada em nuvem Pública.

8- Implementação da solução local:

Exemplo de playbook a ser utilizado no Ansible:

```
- hosts: wordpress
  remote_user: gustavo
```


become: yes

roles:

- server
- php
- mysql
- wordpress

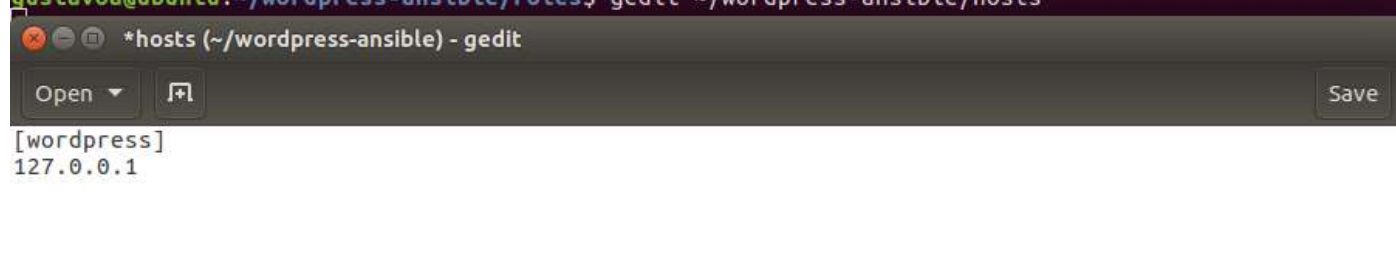
Diretório criado utilizando o Ansible Galaxy:

```
gustavoa@ubuntu:~/playbooks$ cd..
cd..: command not found
gustavoa@ubuntu:~/playbooks$ cd ..
gustavoa@ubuntu:~$ ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  playbooks  Public  Templates  Videos
gustavoa@ubuntu:~$ mkdir wordpress-ansible
gustavoa@ubuntu:~$ cd wordpress-ansible
gustavoa@ubuntu:~/wordpress-ansible$ touch playbook.yml
gustavoa@ubuntu:~/wordpress-ansible$ touch hosts
gustavoa@ubuntu:~/wordpress-ansible$ mkdir roles
gustavoa@ubuntu:~/wordpress-ansible$ cd roles
gustavoa@ubuntu:~/wordpress-ansible/roles$ ansible-galaxy init server
- server was created successfully
gustavoa@ubuntu:~/wordpress-ansible/roles$ ansible-galaxy init php
- php was created successfully
gustavoa@ubuntu:~/wordpress-ansible/roles$ ansible-galaxy init mysql
- mysql was created successfully
gustavoa@ubuntu:~/wordpress-ansible/roles$
```

Figura 01 – Utilizando o Ansible Galaxy

Playbox 100% funcional e implementado no Ansible:

```
gustavoa@ubuntu:~/wordpress-ansible/roles$ gedit ~/wordpress-ansible/hosts
```




The screenshot shows the gedit editor window titled '*hosts (~/.wordpress-ansible) - gedit'. The content of the file is:

```
[wordpress]
127.0.0.1
```

Figura 02 – Preenchendo o arquivo hosts:

```
File Edit View Search Tools Documents Help
gustavoa@ubuntu:~/wordpress-ansible/roles$ cd ~/wordpress-ansible
gustavoa@ubuntu:~/wordpress-ansible$ gedit playbook.yml
```



The screenshot shows the gedit editor window titled '*playbook.yml (~/.wordpress-ansible) - gedit'. The content of the file is:

```
- hosts: wordpress
  remote_user: gustavoa

  become: yes

  roles:
    - server
    - php
    - mysql
    - wordpress
```

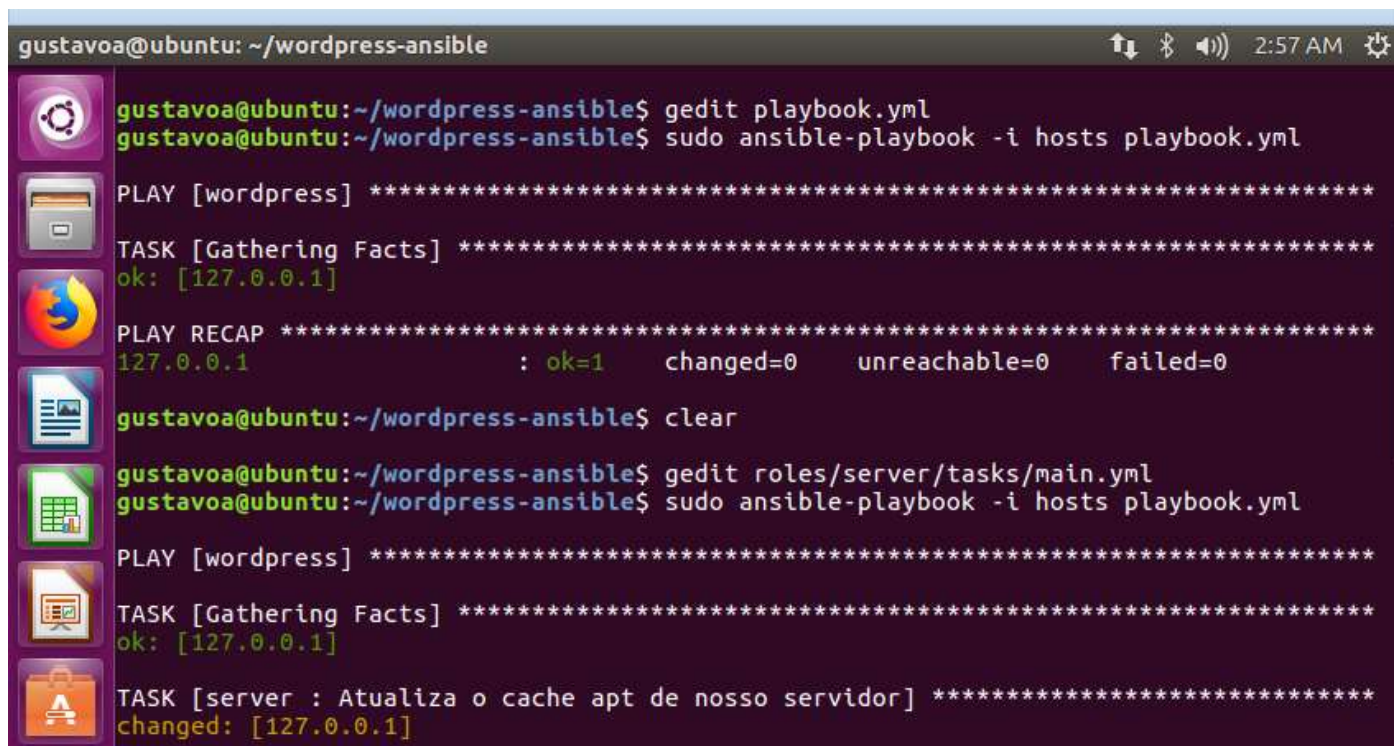
Figura 03 – Preenchendo o playbook:

Criação do repositório para utilização do GitHub:



Figura 03 - <https://github.com/gustavoamaral/ProjblocoTP03.git>

Demonstração do Commit e Push do código:



```

gustavoa@ubuntu: ~/wordpress-ansible
gustavoa@ubuntu:~/wordpress-ansible$ gedit playbook.yml
gustavoa@ubuntu:~/wordpress-ansible$ sudo ansible-playbook -i hosts playbook.yml

PLAY [wordpress] *****
TASK [Gathering Facts] *****
ok: [127.0.0.1]

PLAY RECAP *****
127.0.0.1 : ok=1 changed=0 unreachable=0 failed=0

gustavoa@ubuntu:~/wordpress-ansible$ clear

gustavoa@ubuntu:~/wordpress-ansible$ gedit roles/server/tasks/main.yml
gustavoa@ubuntu:~/wordpress-ansible$ sudo ansible-playbook -i hosts playbook.yml

PLAY [wordpress] *****
TASK [Gathering Facts] *****
ok: [127.0.0.1]

TASK [server : Atualiza o cache apt de nosso servidor] *****
changed: [127.0.0.1]

```

Figura 04 – Código em funcionamento

```

gustavoa@ubuntu:~/wordpress-ansible$ git push -u origin master
Username for 'https://github.com': gustavoaamaral
Password for 'https://gustavoaamaral@github.com':
Counting objects: 54, done.
Compressing objects: 100% (24/24), done.
Writing objects: 100% (54/54), 5.50 KiB | 0 bytes/s, done.
Total 54 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/gustavoaamaral/ProjblocoTP03.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
gustavoa@ubuntu:~/wordpress-ansible$

```

Figura 05 – Push funcionando

```

gustavoa@ubuntu:~/wordpress-ansible$ git commit -m "importando playbook WordPress"
[master (root-commit) 337ec62] importando playbook WordPress
35 files changed, 499 insertions(+)
create mode 100755 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
create mode 100644 roles/server/defaults/main.yml
create mode 100644 roles/server/handlers/main.yml
create mode 100644 roles/server/meta/main.yml
create mode 100644 roles/server/tasks/main.yml
create mode 100644 roles/server/tests/inventory
create mode 100644 roles/server/tests/test.yml
create mode 100644 roles/server/vars/main.yml
create mode 100644 roles/wordpress/README.md
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

```

Figura 06 – Commit funcionando

Relatório de instalação:

- a. Etapa 01 – Criar a pasta wordpress-ansible:


```
# mkdir wordpress-ansible
```

```
# cd wordpress-ansible
```
- b. Etapa 02 – Criar o arquivo “playbook” e o arquivo “hosts” dentro do diretório wordpress-ansible:


```
# touch playbook.yml
```

```
# touch hosts
```
- c. Etapa 03 – Criar as “roles” utilizando o Ansible Galaxy:


```
# ansible-galaxy init server
```

```
# ansible-galaxy init php
```

```
# ansible-galaxy init mysql
```

```
# ansible-galaxy init wordpress
```
- d. Etapa 04 – Configurar o arquivo “hosts”:

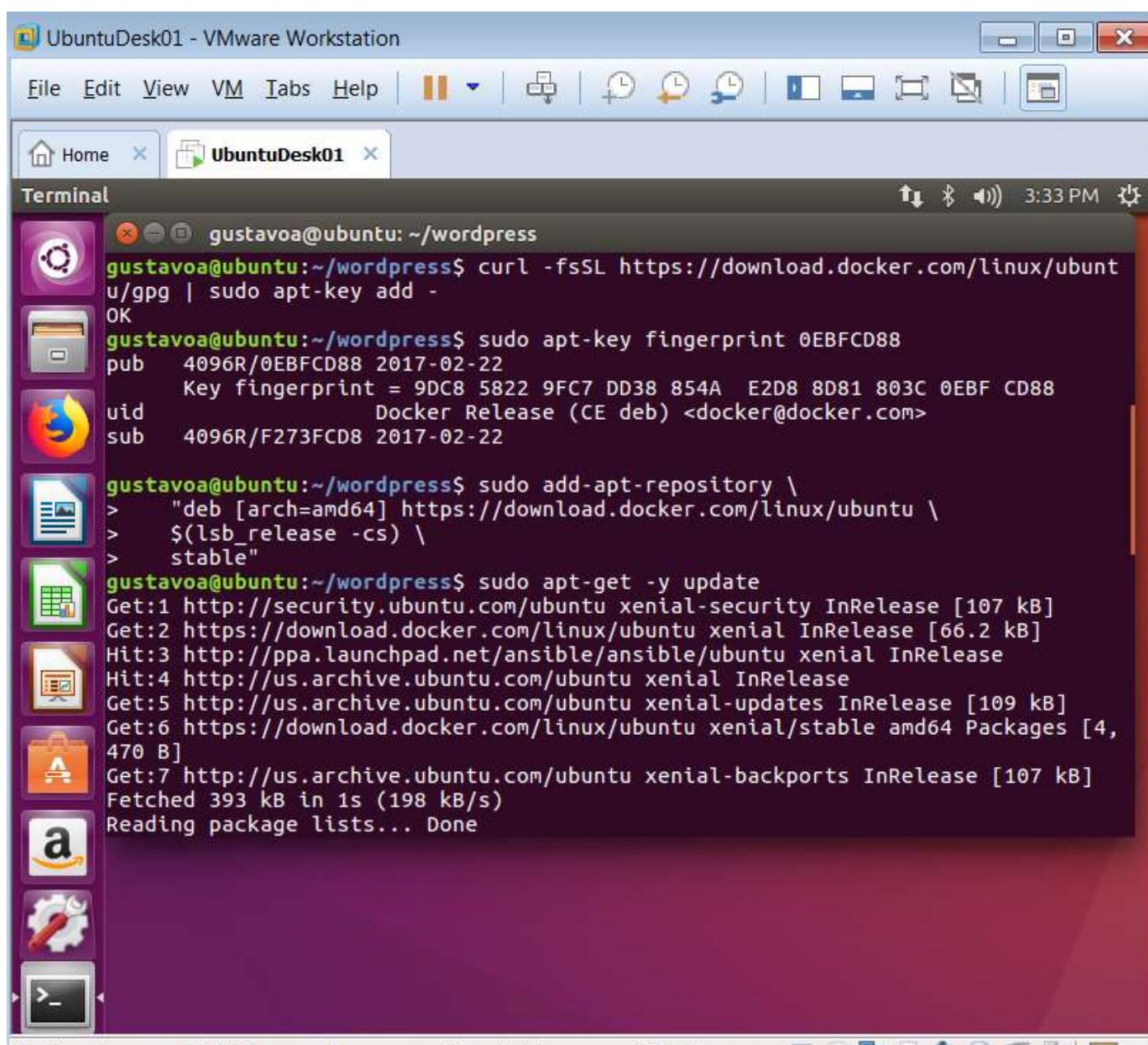

```
# gedit ~/wordpress-ansible/hosts
```

- e. Etapa 05 – Dentro do arquivo “hosts” inserir:
[wordpress]
127.0.0.1
- f. Etapa 06 – Inserindo os comandos dentro do playbook do Ansible:
cd ~/wordpress-ansible
gedit playbook.yml
- g. Etapa 07 – Conteúdo do playbook do Ansible:
- hosts: wordpress
 remote_user: gustavo

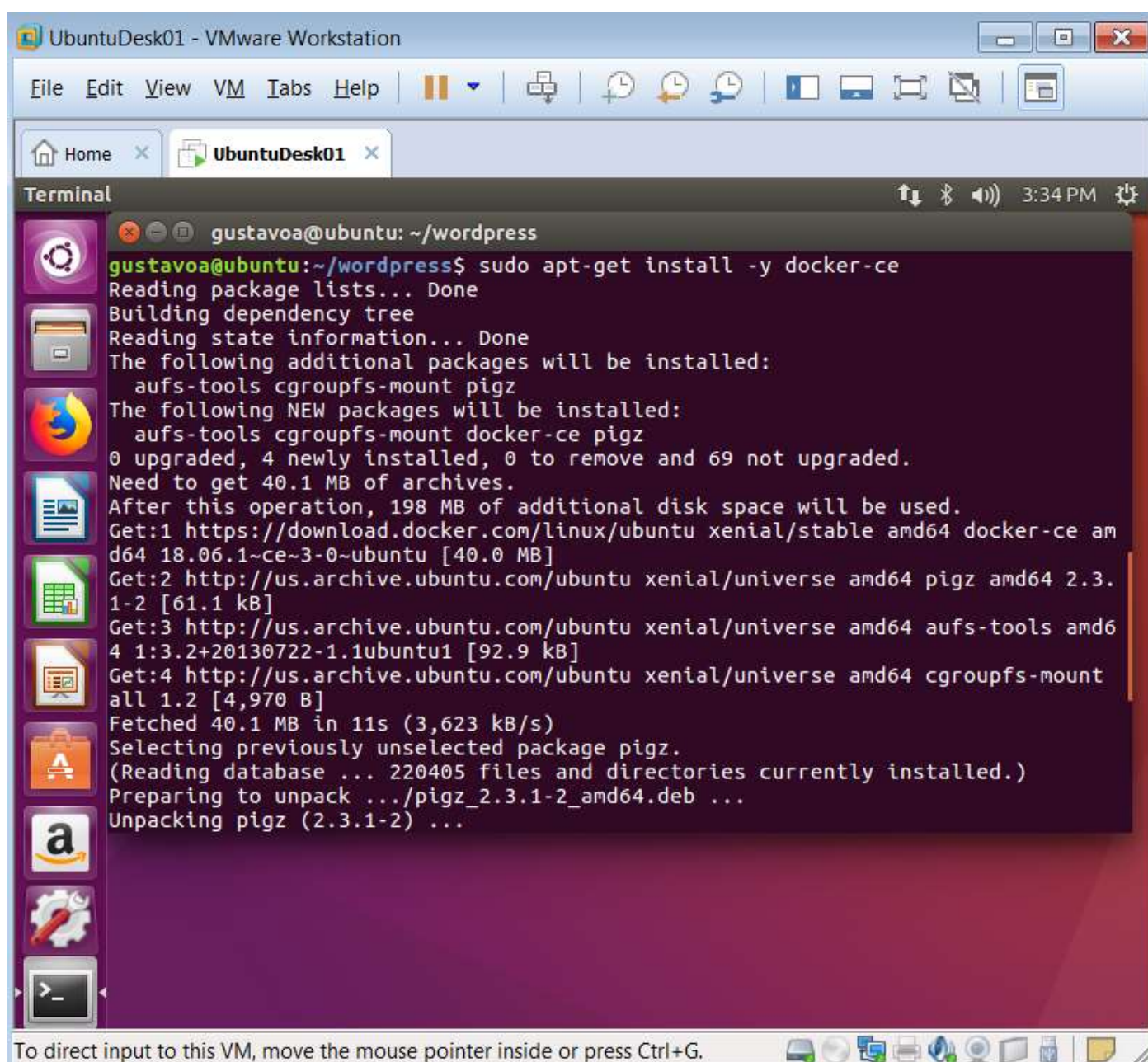
 become: yes

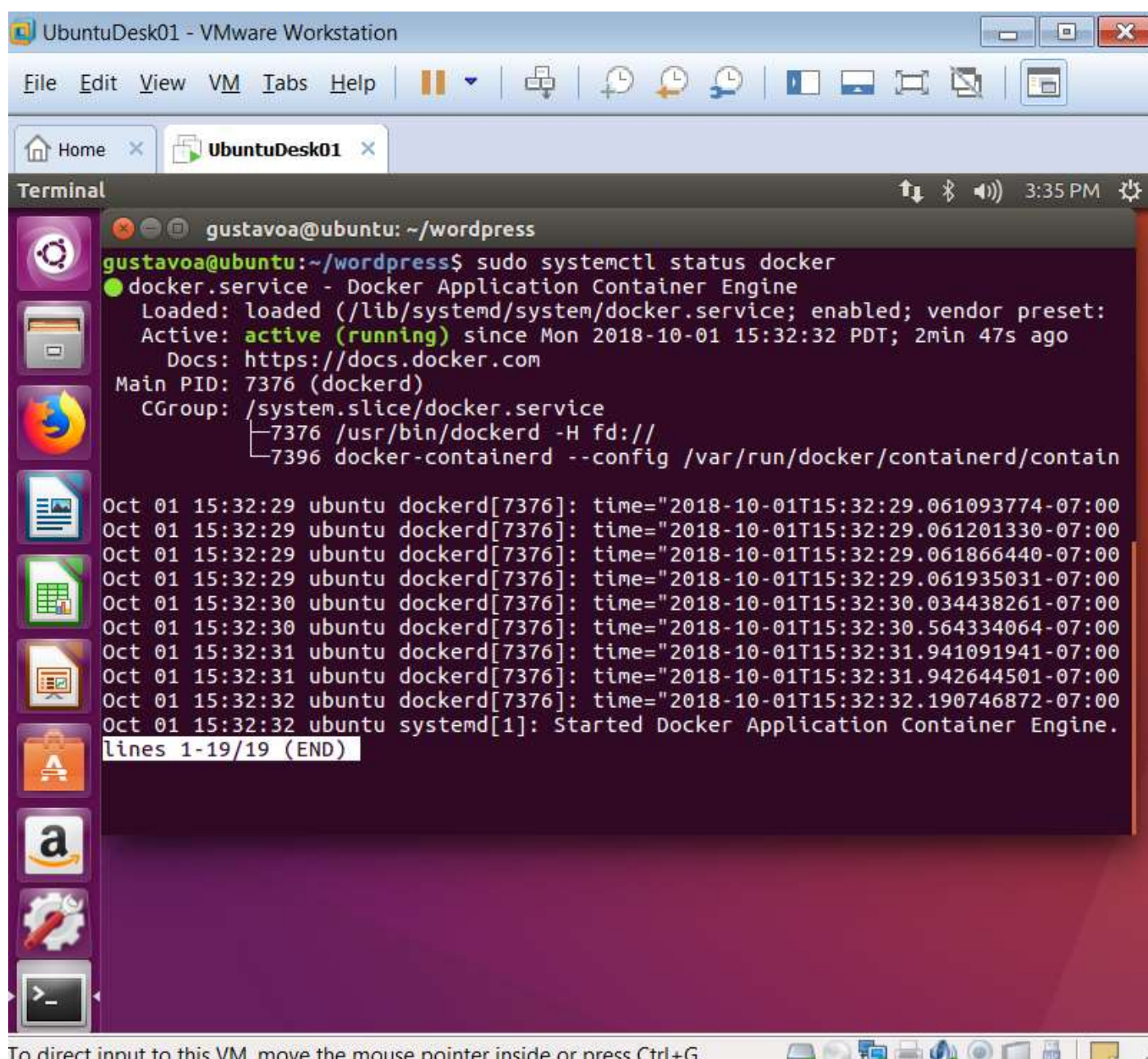
 roles:
 - server
 - php
 - mysql
 - wordpress
- h. Etapa 08 – Execução do playbook para primeiro teste:
ansible-playbook -i hosts playbook.yml

Detalhamento da instalação do Docker Compose:



```
gustavoa@ubuntu: ~/wordpress
gustavoa@ubuntu:~/wordpress$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
gustavoa@ubuntu:~/wordpress$ sudo apt-key fingerprint 0EBFCD88
pub 4096R/0EBFCD88 2017-02-22
Key fingerprint = 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid Docker Release (CE deb) <docker@docker.com>
sub 4096R/F273FCD8 2017-02-22
gustavoa@ubuntu:~/wordpress$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
gustavoa@ubuntu:~/wordpress$ sudo apt-get -y update
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [107 kB]
Get:2 https://download.docker.com/linux/ubuntu xenial InRelease [66.2 kB]
Hit:3 http://ppa.launchpad.net/ansible/ansible/ubuntu xenial InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:5 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:6 https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages [4,470 B]
Get:7 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Fetched 393 kB in 1s (198 kB/s)
Reading package lists... Done
```





```
gustavoa@ubuntu: ~/wordpress
gustavoa@ubuntu:~/wordpress$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Mon 2018-10-01 15:32:32 PDT; 2min 47s ago
     Docs: https://docs.docker.com
   Main PID: 7376 (dockerd)
    CGroup: /system.slice/docker.service
            └─7376 /usr/bin/dockerd -H fd://
               └─7396 docker-containerd --config /var/run/docker/containerd/contain

Oct 01 15:32:29 ubuntu dockerd[7376]: time="2018-10-01T15:32:29.061093774-07:00
Oct 01 15:32:29 ubuntu dockerd[7376]: time="2018-10-01T15:32:29.061201330-07:00
Oct 01 15:32:29 ubuntu dockerd[7376]: time="2018-10-01T15:32:29.061866440-07:00
Oct 01 15:32:29 ubuntu dockerd[7376]: time="2018-10-01T15:32:29.061935031-07:00
Oct 01 15:32:30 ubuntu dockerd[7376]: time="2018-10-01T15:32:30.034438261-07:00
Oct 01 15:32:30 ubuntu dockerd[7376]: time="2018-10-01T15:32:30.564334064-07:00
Oct 01 15:32:31 ubuntu dockerd[7376]: time="2018-10-01T15:32:31.941091941-07:00
Oct 01 15:32:31 ubuntu dockerd[7376]: time="2018-10-01T15:32:31.942644501-07:00
Oct 01 15:32:32 ubuntu dockerd[7376]: time="2018-10-01T15:32:32.190746872-07:00
Oct 01 15:32:32 ubuntu systemd[1]: Started Docker Application Container Engine.
lines 1-19/19 (END)
```

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

UbuntuDesk01 - VMware Workstation

File Edit View VM Tabs Help

Home x UbuntuDesk01 x

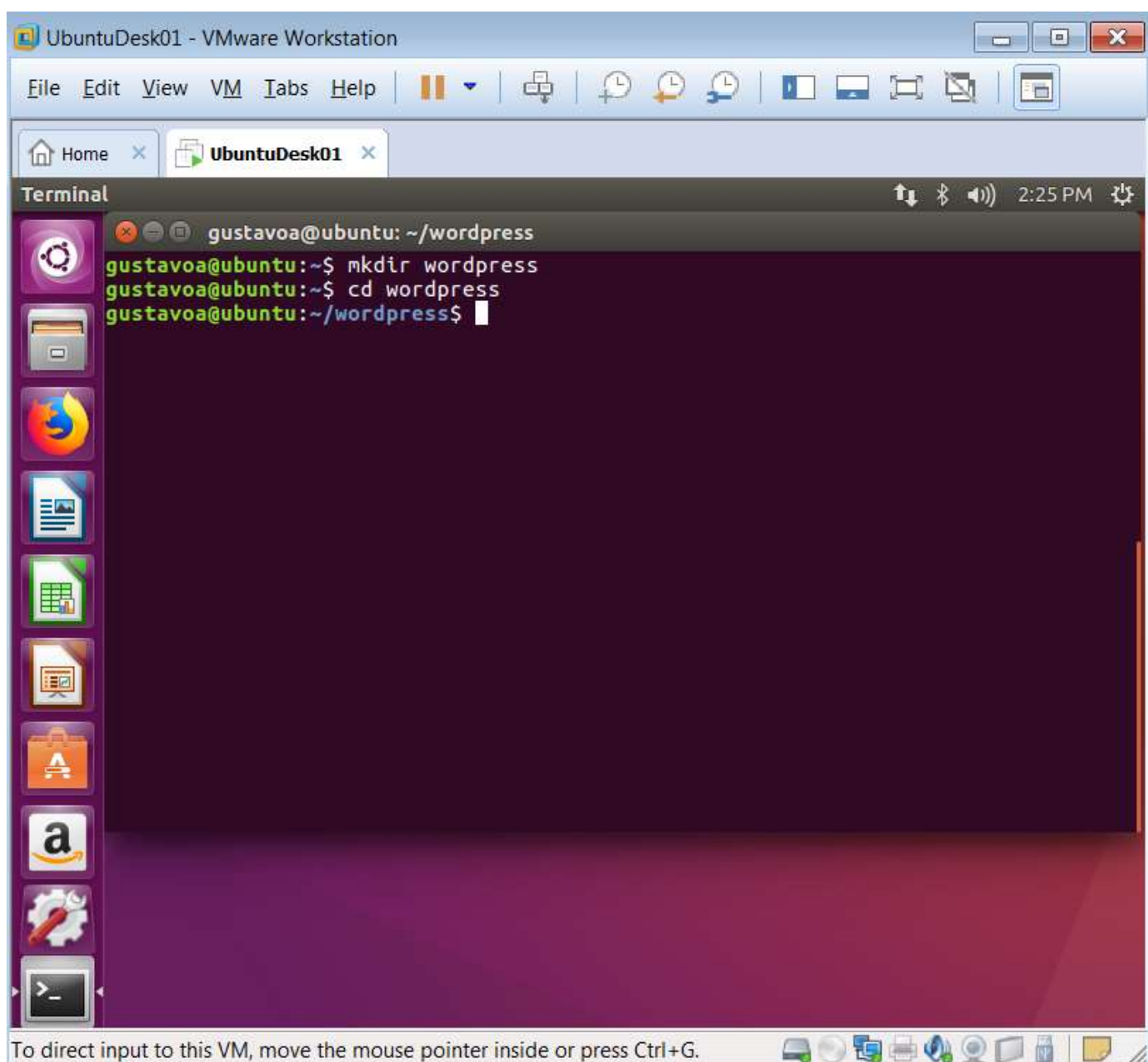
Terminal 2:16 PM

```
gustavoa@ubuntu: ~
gustavoa@ubuntu:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.22.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 617    0 617    0     0    268      0 --:--:--  0:00:02 --:--:--  268
100 11.2M 100 11.2M    0     0 1658k      0 0:00:06  0:00:06 --:--:-- 3141k
gustavoa@ubuntu:~$ sudo chmod +x /usr/local/bin/docker-compose
gustavoa@ubuntu:~$
```

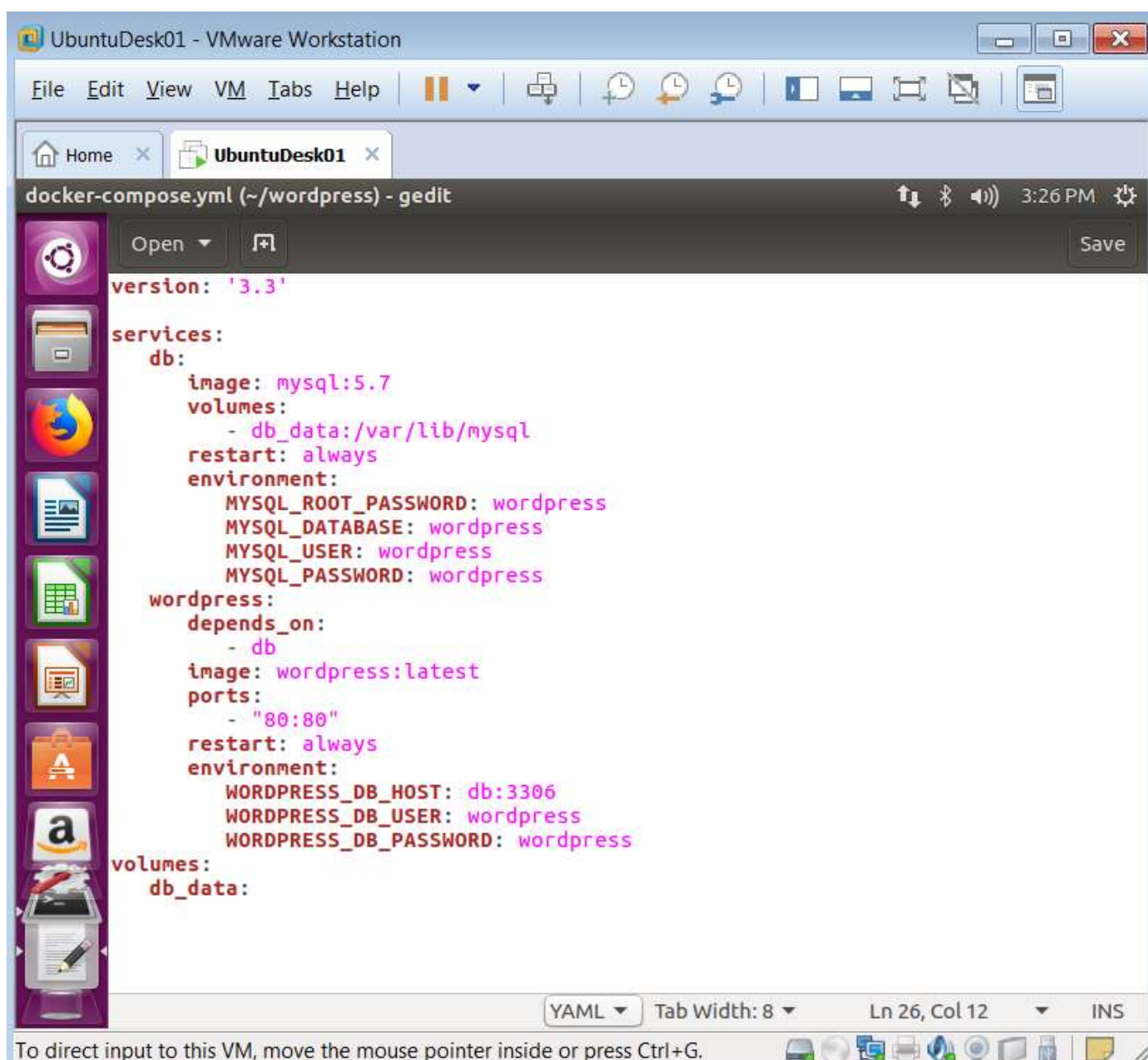
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

Comandos necessários para a execução dos containers:

Criação de diretório Docker (onde ficará armazenado os arquivos Docker-Compose.yml – responsáveis pelas configurações dos containers):



**Criação do arquivo Docker-compose.yml (importante: não usar tab na endentação!
Utilizar espaço)**

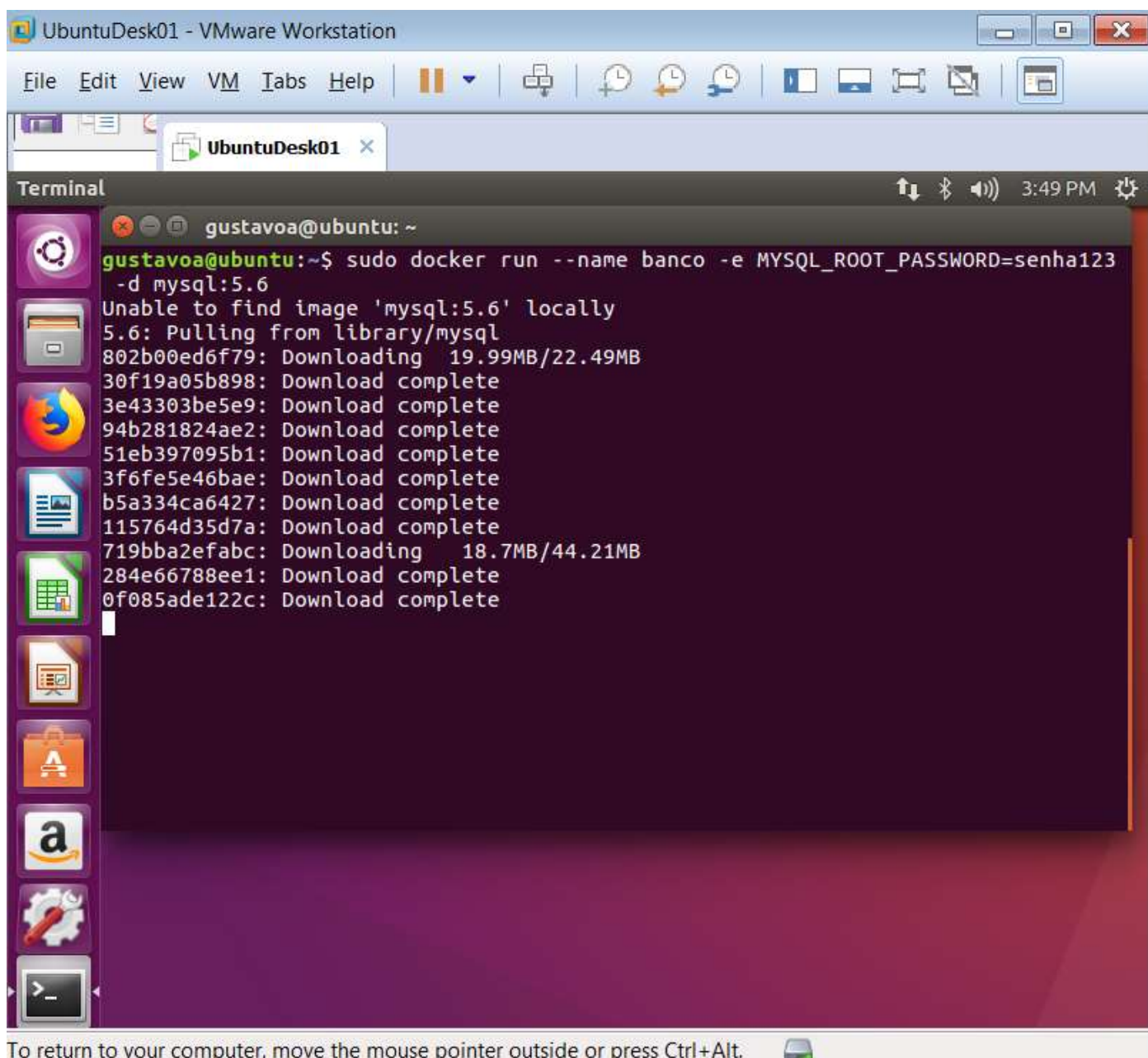


```
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: wordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
volumes:
  db_data:
```

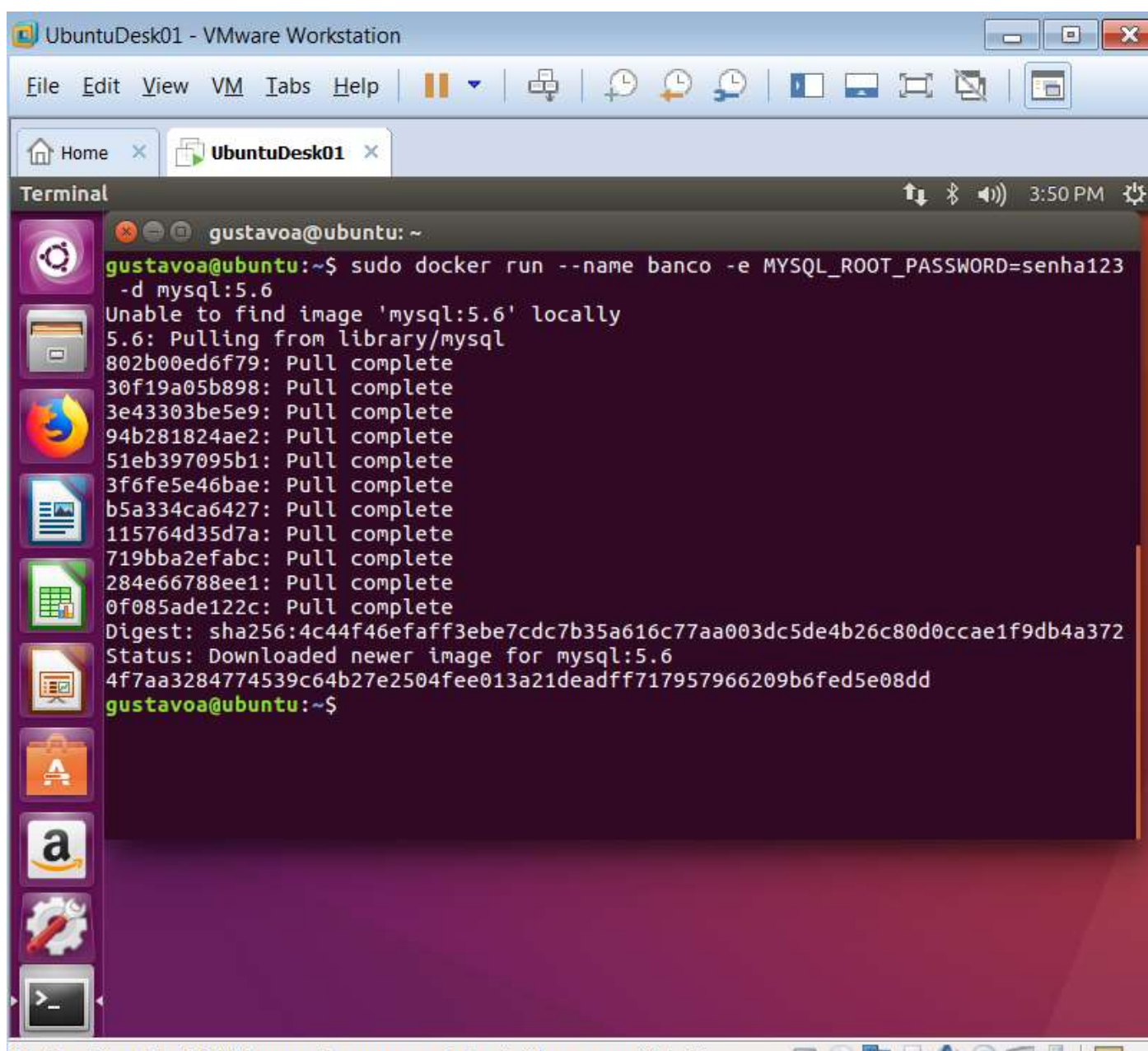
Ativação dos containers (Banco de dados MySQL e Wordpress)

Comando: `$ sudo docker run --name banco -e MYSQL_ROOT_PASSWORD=senha123 -d mysql:5.6`

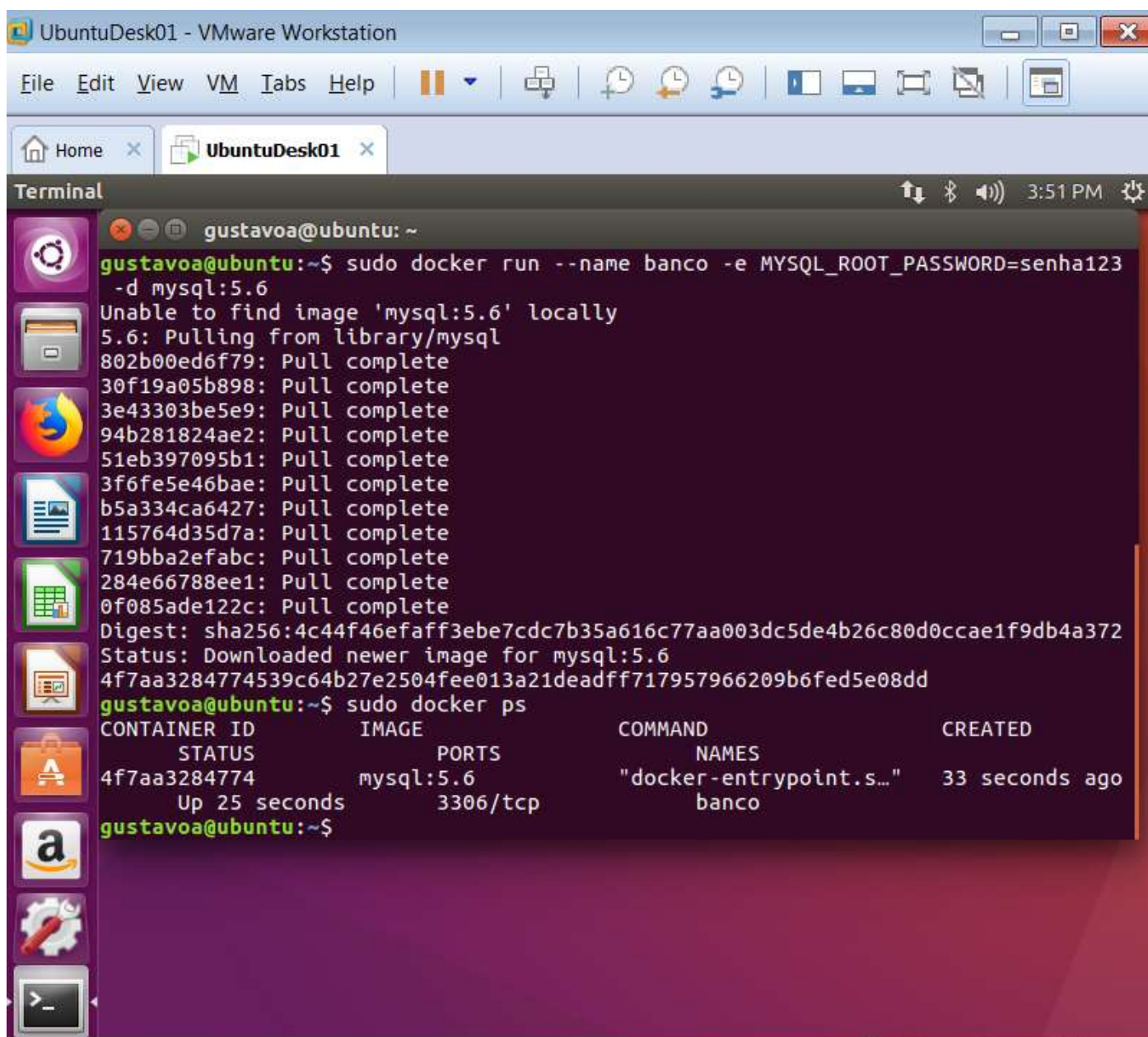


```
gustavoa@ubuntu: ~  
gustavoa@ubuntu:~$ sudo docker run --name banco -e MYSQL_ROOT_PASSWORD=senha123  
-d mysql:5.6  
Unable to find image 'mysql:5.6' locally  
5.6: Pulling from library/mysql  
802b00ed6f79: Downloading 19.99MB/22.49MB  
30f19a05b898: Download complete  
3e43303be5e9: Download complete  
94b281824ae2: Download complete  
51eb397095b1: Download complete  
3f6fe5e46bae: Download complete  
b5a334ca6427: Download complete  
115764d35d7a: Download complete  
719bba2efabc: Downloading 18.7MB/44.21MB  
284e66788ee1: Download complete  
0f085ade122c: Download complete
```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.



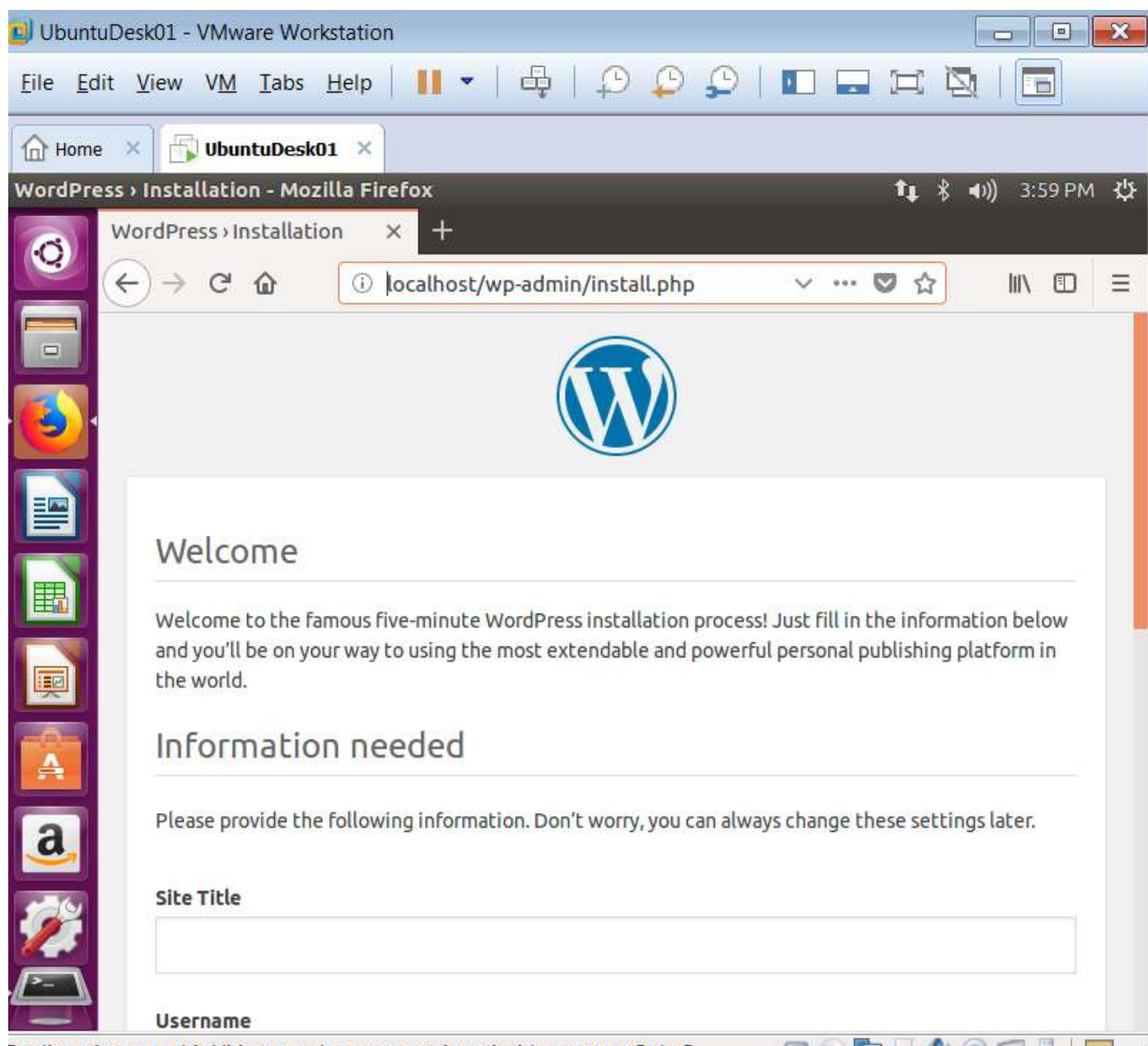
Container em execução – Comando: `$ sudo docker ps`



```
gustavoa@ubuntu: ~$ sudo docker run --name banco -e MYSQL_ROOT_PASSWORD=senha123
-d mysql:5.6
Unable to find image 'mysql:5.6' locally
5.6: Pulling from library/mysql
802b00ed6f79: Pull complete
30f19a05b898: Pull complete
3e43303be5e9: Pull complete
94b281824ae2: Pull complete
51eb397095b1: Pull complete
3f6fe5e46bae: Pull complete
b5a334ca6427: Pull complete
115764d35d7a: Pull complete
719bba2efabc: Pull complete
284e66788ee1: Pull complete
0f085ade122c: Pull complete
Digest: sha256:4c44f46efaff3ebe7cdc7b35a616c77aa003dc5de4b26c80d0ccae1f9db4a372
Status: Downloaded newer image for mysql:5.6
4f7aa3284774539c64b27e2504fee013a21deadff717957966209b6fed5e08dd
gustavoa@ubuntu: ~$ sudo docker ps
CONTAINER ID        IMAGE               PORTS              COMMAND              NAMES              CREATED
STATUS              PORTS              NAMES
4f7aa3284774        mysql:5.6          3306/tcp           "docker-entrypoint.s...  banco              33 seconds ago
Up 25 seconds
```

Comando: `$ sudo docker run --name meusite --link banco:mysql -p 80:80 -d wordpress`

Containers estavam acessíveis por um cliente (print de telas):



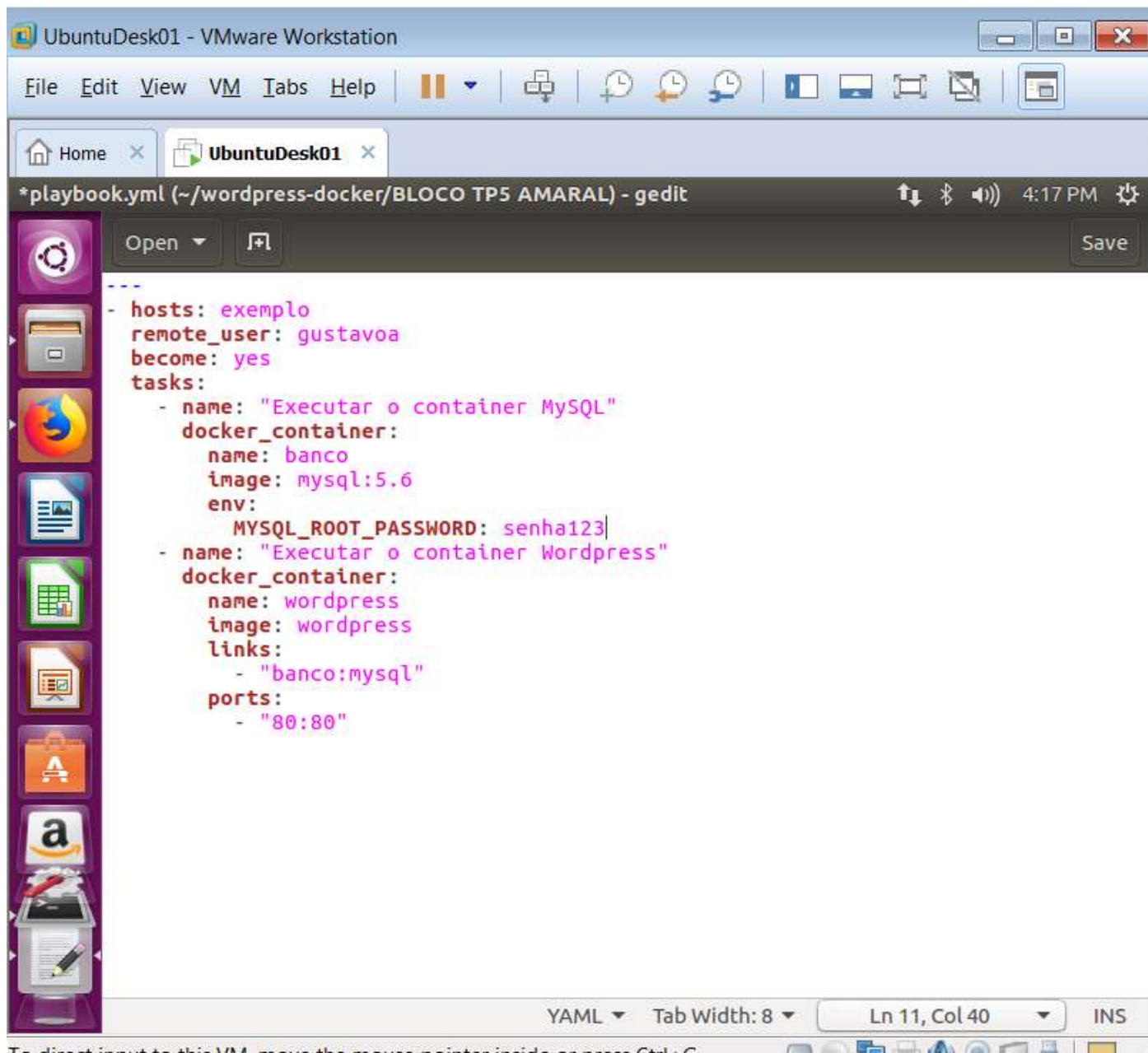
Playbook Ansible:

```

---
- hosts: SEUGRUPO
  remote_user: SEULOGIN
  become: yes
  tasks:
    - name: "Executa o container MySQL"
      docker_container:
        name: banco
        image: mysql:5.6
        env:
          MYSQL_ROOT_PASSWORD: senha123
    - name: "Executa o container WordPress"
      docker_container:
        name: wordpress
        image: wordpress
        links:
  
```


- "banco:mysql"
- ports:
- "80:80"

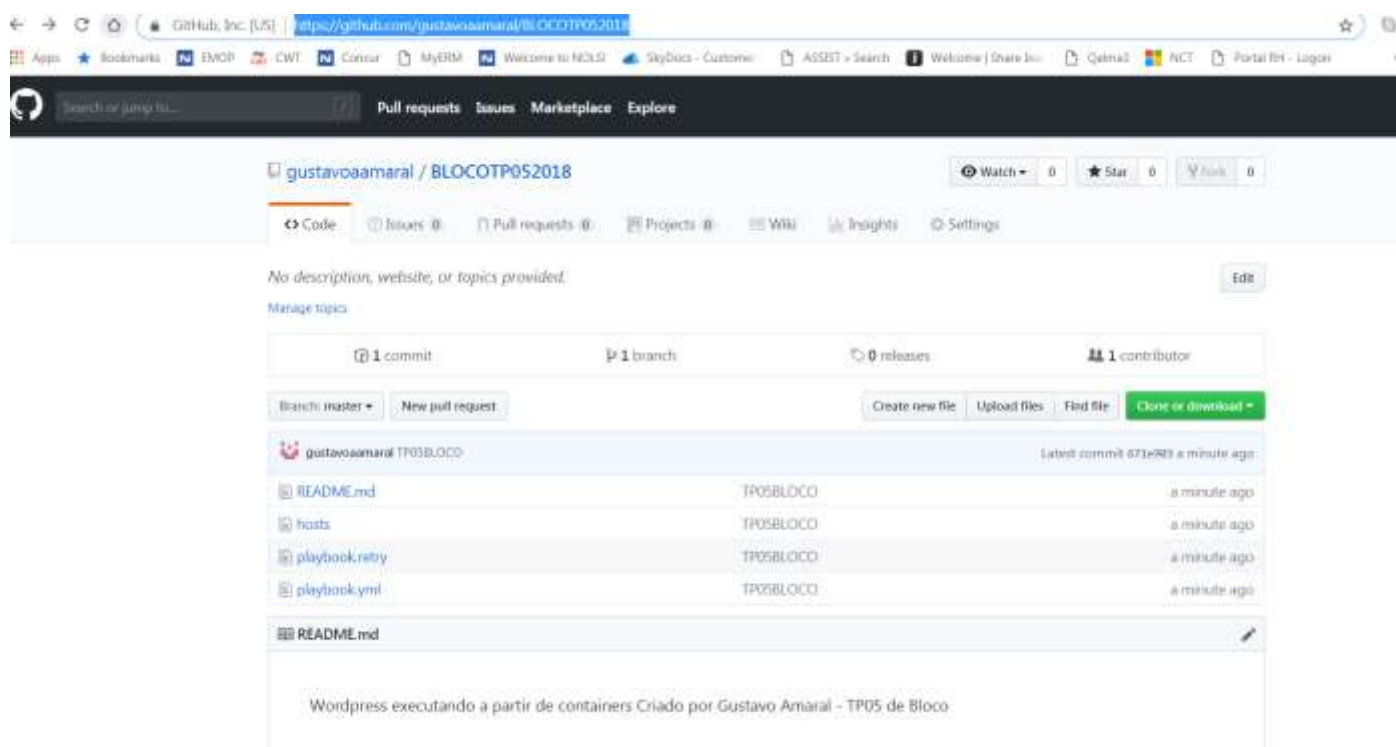
Demonstração do playbook ansible funcional:



```
---
- hosts: exemplo
  remote_user: gustavo
  become: yes
  tasks:
    - name: "Executar o container MySQL"
      docker_container:
        name: banco
        image: mysql:5.6
        env:
          MYSQL_ROOT_PASSWORD: senha123
    - name: "Executar o container Wordpress"
      docker_container:
        name: wordpress
        image: wordpress
        links:
          - "banco:mysql"
        ports:
          - "80:80"
```

Playbook Ansible publicado no repositório Github:

<https://github.com/gustavoamaral/BLOCOTP052018>



9- Implementação da solução em nuvem Pública:

a. Detalhamento dos serviços disponíveis na Amazon AWS:

i. AWS:

Amazon Web Services, também conhecido como AWS, é uma plataforma de serviços de computação em nuvem, que formam uma plataforma de computação na nuvem oferecida pela Amazon.com. Os serviços são oferecidos em 55 diferentes zonas distribuídas em 18 regiões geográficas no mundo.

ii. EC2:

Amazon Elastic Compute Cloud é uma parte central da plataforma de cloud computing da Amazon.com, Amazon Web Services. O EC2 permite que os usuários aluguem computadores virtuais nos quais rodam suas próprias aplicações.

iii. VPC:

O Amazon Virtual Private Cloud é um serviço de computação em nuvem comercial que fornece aos usuários uma nuvem virtual privada, "provisionando uma seção isolada logicamente da Amazon Web Services Cloud".

iv. Elastic Load Balance:

O Elastic Load Balancing distribui ou tráfego de rede de entrada dos aplicativos em múltiplos destinos, como instâncias do Amazon EC2,

contêineres e endereços IP, em várias zonas de disponibilidade. O Elastic Load Balancing escala o load balancer à medida que o tráfego para seu aplicativo muda com o passar do tempo, e pode escalar para a vasta maioria das cargas de trabalho automaticamente.

v. Route53:

O Amazon Route 53 é um sistema de nomes de domínio escalável e altamente disponível. Lançado em 5 de dezembro de 2010, faz parte da plataforma de computação em nuvem da Amazon.com, Amazon Web Services. O nome é uma referência à porta TCP ou UDP 53, na qual as solicitações do servidor DNS são endereçadas.

vi. S3:

O Amazon S3 ou o Amazon Simple Storage Service é um "serviço de armazenamento simples" oferecido pelo Amazon Web Services que fornece armazenamento de objetos por meio de uma interface de serviço da web.

b. Detalhamento da implantação do serviço na nuvem Pública AWS:

i. Instalação da interface de comando AWS no Ubuntu:

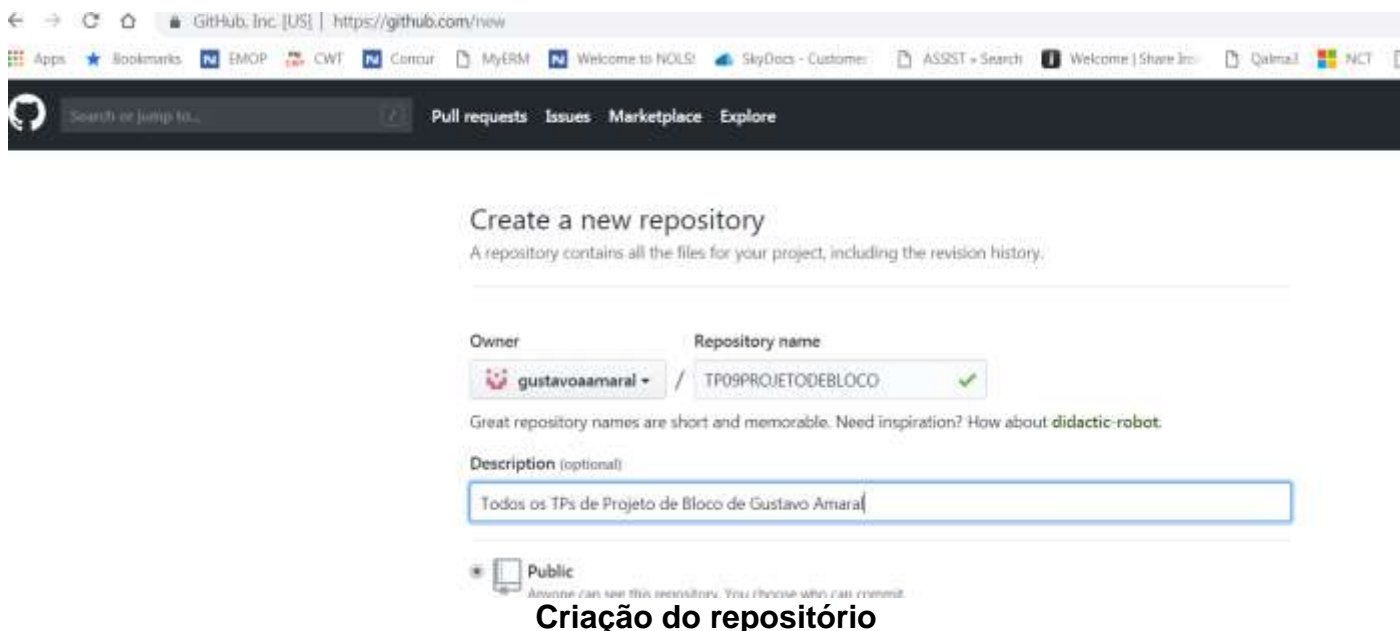
```

gustavoa@ubuntu: ~
d982e3946ac5: Pull complete
a38e2fdf4f50: Pull complete
09f702917a0a: Pull complete
df1d46358537: Pull complete
9b0d5695ec42: Pull complete
7abe1c1f0479: Pull complete
db1df7737fbd: Pull complete
da5248206256: Pull complete
e850a08a7c7e: Pull complete
e5e7ecd1752b: Pull complete
Digest: sha256:7e476394586459bb622d3f37448cd07e703ec6906257d232542f2f51ff073da7
Status: Downloaded newer image for wordpress:latest
e7ef35993b283b6a067c6e9e80965c3cc76125f229e5a285141fdfe4078b0bc8
gustavoa@ubuntu:~$ sudo apt-get install awscli groff
[sudo] password for gustavoa:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docutils-common psutils python3-boto3 python3-colorama python3-dateutil
  python3-docutils python3-jmespath python3-pygments python3-roman python3-rsa
  python3-s3transfer
Suggested packages:
  texlive-latex-recommended texlive-latex-base texlive-lang-french
  fonts-linuxlibertine | ttf-linux-libertine docutils-doc ttf-bitstream-vera
The following NEW packages will be installed:
  awscli docutils-common groff psutils python3-boto3 python3-colorama
  python3-dateutil python3-docutils python3-jmespath python3-pygments
  python3-roman python3-rsa python3-s3transfer
0 upgraded, 13 newly installed, 0 to remove and 174 not upgraded.
Need to get 5,954 kB of archives.
After this operation, 37.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

- ii. Observação – Esses passos serão demonstrados com mais detalhes na apresentação de projeto de blocos

10- Upload dos Todos os TPs no Github:



GitHub, Inc. [US] | <https://github.com/new>

Apps Bookmarks EMOP CWT Coursera MyERM Welcome to NOLSI SkyDocs - Customer ASSIST + Search Welcome | Share | Qalmar NCT

Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: **gustavoamaral** / Repository name: **TP09PROJETODEBLOCO** ✓

Great repository names are short and memorable. Need inspiration? How about `didactic-robot`.

Description (optional): **Todos os TPs de Projeto de Bloco de Gustavo Amaral**

☒ Public
Anyone can see this repository. You choose who can commit.

Criação do repositório

Link: <https://github.com/gustavoamaral/TP09PROJETODEBLOCO>

11-Conclusão:

Com os tópicos propostos neste documento, espera-se que todos objetivos propostos no mesmo, sejam alcançados, e que a implementação desta solução esteja totalmente funcional de acordo com as premissas abordadas.

TP09 – Rubrica (Competência):

TP6

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno caracterizou o problema a ser tratado adequadamente e apresentou uma aplicação realmente distribuída (pelo menos com uma camada de banco de dados e outra de aplicação, por exemplo), capaz de tratar o problema enunciado?

TP6

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou o problema dentro de um contexto adequado, relatando por exemplo uma empresa (pode ser fictícia) em que a solução será implantada, que tipo de processos/atividades/pessoas serão atendidas e que tipo de benefícios ele espera produzir com seu projeto?

TP6

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno fez uma apresentação da aplicação distribuída que pretende implantar, dentro dos requisitos apresentados no item c?

TP6

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou a organização de sua infraestrutura, com pelo menos um diagrama de rede contendo os diferentes elementos de virtualização, detalhes sobre a rede e uma página de texto completa?

TP7

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno fez uma rápida revisão das principais referências sobre as tecnologias abordadas em seu trabalho, informando o tipo de virtualização, edições, recursos e custos aproximados?

TP7

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O texto apresentado pelo aluno conta com pelo menos uma seção textual de uma página comparando a solução de virtualização utilizada no projeto e as demais?

TP7

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou um planejamento da instalação, com os passos a serem realizados, na ordem em que devem acontecer e de forma coerente?

TP7

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou um cronograma coerente com as estimativas de tempo para implantação da solução pretendida?

TP8

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou referências de download e/ou de versões para todos os componentes usados em sua solução?

TP8

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou a sequência e evidências da execução dos passos de implementação da sua infraestrutura de virtualização?

TP8

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou a sequência e evidências da execução dos passos de configuração da aplicação rodando sobre sua infraestrutura virtualizada?

TP8

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou evidências do funcionamento da aplicação que ele pretendia contemplar em sua solução?

1. Propor soluções de arquitetura de infraestrutura para aplicações de acordo com o cenário

O aluno incluiu em seu trabalho informações sobre como a solução a ser implantada recebe o input de seus usuários. Estas informações são coerentes e refletem as características da aplicação proposta?

TP9

1. Propor soluções de arquitetura de infraestrutura para aplicações de acordo com o cenário

O aluno propôs uma arquitetura que está de acordo com o modelo SDDC?

TP9

1. Propor soluções de arquitetura de infraestrutura para aplicações de acordo com o cenário

O aluno conseguiu levantar os detalhes de como o código-fonte da aplicação a ser implantada é gerenciado (sistema de versionamento, local de hospedagem, etc.)?

TP9

1. Propor soluções de arquitetura de infraestrutura para aplicações de acordo com o cenário

Os passos elencados pelo aluno para transformar o código da aplicação em um pacote instalável são coerentes? Não é necessário fornecer dados muito específicos, como comandos de compilação ou resultados, mas é preciso informar

itens de alto nível como a linguagem a ser usada, e/ou a estratégia usada para baixar e preparar o software para instalação. Se ele não tem código-fonte disponível, os detalhes gerais do processo de download e instalação são suficientes.

TP9

2. Utilizar ferramentas de automação de instalação de aplicações

O aluno conseguiu automatizar alguma parte do processo de implantação da aplicação?

TP9

2. Utilizar ferramentas de automação de instalação de aplicações

O aluno se preocupou em tornar os elementos de sua implantação reutilizáveis, de forma que outras instalações baseadas na mesma solução possam reutilizar código e/ou comandos usados no trabalho?

TP9

2. Utilizar ferramentas de automação de instalação de aplicações

O aluno se preocupou em consultar ou implementar as melhores práticas das soluções usadas, como por exemplo usando datastores adequados em um ambiente VMware, ou usando AMIs oficiais no caso da AWS (ao invés de ambientes pré-instalados de terceiros)?

TP9

2. Utilizar ferramentas de automação de instalação de aplicações

O aluno usou os recursos de automação adotados ao longo de seu trabalho?

TP9

3. Documentar um projeto de infraestrutura para aplicações

O aluno criou o repositório indicado no enunciado?

TP9

3. Documentar um projeto de infraestrutura para aplicações

O aluno conseguiu fazer o commit e o push do seu trabalho corretamente?

TP9

3. Documentar um projeto de infraestrutura para aplicações

Todos os recursos usados no trabalho (scripts, playbooks, programas, se houverem) estão armazenados em um repositório Git?

TP9

3. Documentar um projeto de infraestrutura para aplicações

A URL de repositório fornecida pelo aluno tem um conteúdo consistente com o relatório?

TP9

4. Levantar requisitos para um projeto de infraestrutura para aplicações

O aluno escolheu uma das plataformas apresentadas ao longo do bloco para implementar sua aplicação, e o fez apropriadamente?

TP9

4. Levantar requisitos para um projeto de infraestrutura para aplicações

O aluno conseguiu executar a aplicação pretendida com sucesso?

TP9

4. Levantar requisitos para um projeto de infraestrutura para aplicações

Os passos necessários para a instalação da aplicação apresentados no texto do projeto estão consistentes com a execução documentada e apresentada?

TP9

4. Levantar requisitos para um projeto de infraestrutura para aplicações

O aluno conseguiu demonstrar o acesso de um cliente à aplicação, através das capturas de tela apresentadas?

TP9

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno caracterizou o problema a ser tratado adequadamente e apresentou uma aplicação realmente distribuída (pelo menos com uma camada de banco de dados e outra de aplicação, por exemplo), capaz de tratar o problema enunciado?

TP9

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno fez uma rápida revisão das principais referências sobre as tecnologias abordadas em seu trabalho, informando o tipo de virtualização, edições, recursos e custos aproximados?

TP9

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno apresentou a sequência e evidências da execução dos passos de implementação da sua infraestrutura de virtualização?

TP9

5. Elaborar uma solução de infraestrutura para aplicações com base em requisitos

O aluno fez ao menos uma sugestão de melhoria, realista, para a implantação que apresentou?

TP09 – Referências :

Acesso em: 20/11/2018

- <https://lms.infnet.edu.br/moodle/mod/assign/view.php?id=94776>
- <https://lms.infnet.edu.br/moodle/mod/assign/view.php?id=94768>
- <https://lms.infnet.edu.br/moodle/mod/page/view.php?id=94774>
- <https://lms.infnet.edu.br/moodle/mod/page/view.php?id=94766>
- <https://lms.infnet.edu.br/moodle/course/view.php?id=1601>
- <https://wordpress.org/about/requirements/>
- <https://www.openstack.org/assets/software/mitaka/OpenStack-WorkloadRefArchWebApps-v7.pdf>
- <https://www.rackspace.com/pt/calculator>
- https://aws.amazon.com/pt/what-is-cloud-computing/?sc_channel=PS&sc_campaign=acquisition_BR&sc_publisher=google&sc_medium=br_cloud_computing_b&sc_content=sitelink&sc_detail=%2Bamazon%20%2Bnuvens&sc_category=cloud_computing&sc_segment=what_is_cloud_computing&sc_matchtype=b&sc_country=BR&sc_kwcid=AL!4422!3!163931251575!b!l!%2Bamazon%20%2Bnuvens&ef_id=W711xQAAAPo0ItS X:20181021003958:s
- <https://www.canalwp.com/tutorial-wordpress/criar-banco-de-dados-wordpress-cpanel/>
- <https://canaltech.com.br/internet/O-que-e-servidor-Apache/>
- https://aws.amazon.com/pt/what-is-cloud-computing/?sc_channel=PS&sc_campaign=acquisition_BR&sc_publisher=google&sc_medium=br_cloud_computing_b&sc_content=sitelink&sc_detail=%2Bamazon%20%2Bnuvens&sc_category=cloud_computing&sc_segment=what_is_cloud_computing&sc_matchtype=b&sc_country=BR&sc_kwcid=AL!4422!3!163931251575!b!l!%2Bamazon%20%2Bnuvens&ef_id=W711xQAAAPo0ItS X:20181021003958:s
- <https://aws.amazon.com/pt/products/>
- <http://harish11g.blogspot.com/2013/03/Configuring-installing-Apache-Solrcloud-solr4.x-on-Amazon-VPC-EC2-AWS.html>
- https://docs.aws.amazon.com/pt_br/AmazonECS/latest/developerguide/Welcome.html
- <https://uksysadmin.wordpress.com/2011/02/24/running-openstack-under-virtualbox-a-complete-guide-part-2/>
- <https://awstccalculator.com/#>
- <https://aws.amazon.com/pt/choosing-a-cloud-platform/>
- <https://tecnoblog.net/241060/ubuntu-18-04-lts-gnome/>
- <https://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>
- <https://www.hostinger.com.br/tutoriais/o-que-e-apache#gref>
- <https://canaltech.com.br/internet/O-que-e-servidor-Apache/>
- https://www.redhat.com/pt-br/engage/delivery-with-ansible-20170906?sc_cid=701f2000000tvVIAAI&gclid=Cj0KCQiA_4jgBRDhARIsADeZxcj_nAYTOXoft4E3Nr9GN2a11EXMdXWWm3vPi6dbWuBfhXTxOiRx4y8aAuOhEALw_wcB&gclsrc=aw.ds
- <https://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/>
- <https://www.oficinadanet.com.br/post/16111-o-que-e-e-como-funciona-o-git-e-github>
- https://www.hostmidia.com.br/blog/o-que-e-wordpress/?utm_source=google&utm_medium=cpc&utm_campaign=oqewp&gclid=Cj0KCQiA_4jgBRDhARIsADeZxch5U6pyNI6b2YKqUKUH_32tSBIOyJlJaTrnIPUA7qkwHbOj4TrDGvwaAhkXEALw_wcB
- <https://www.mundodocker.com.br/o-que-e-docker/>
- <https://aws.amazon.com/pt/ec2/vm-import/>
- https://docs.aws.amazon.com/pt_br/vm-import/latest/userguide/what-is-vmimport.html
- https://docs.aws.amazon.com/pt_br/cli/latest/userguide/cli-chap-welcome.html
- <https://giovannireisnunes.wordpress.com/2016/09/02/primeiros-passos-em-aws-e-aws-cli/>

