

# **GRADUAÇÃO EM REDE DE COMPUTADORES**



*Projeto de Bloco - Arquitetura e Infraestrutura de Aplicações [18E3\_5] – Teste de Performance – TP05*

*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  
22/09/2018*

## Sumário

TP05 – Introdução:.....	3
Descrição .....	3
TP05 – Parte 1: .....	3
TP05 – Rubrica (Competência):.....	17
TP05 – Referências : .....	18

## TP05 – 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.

##### TP05

Apresente um playbook Ansible que execute uma aplicação distribuída, composta por dois ou mais containers Docker ou VMs em um host vSphere. Seu playbook deve ser disponibilizado em um repositório GitHub, e seu trabalho deve incluir a URL para o repositório. Seu trabalho também deve apresentar o resultado da execução do playbook.

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

## TP05 – Parte 1:

#### 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 colocar em prática a utilização da ferramenta de automação de rotinas e comandos conhecida como Ansible.

A implementação levará em conta a tecnologia de virtualização baseada em container.

### Desenvolvimento:

#### 1- Escolha da aplicação:

Para implementação de uma estrutura virtualizada em container, foi escolhida a aplicação Docker Compose.

O Docker Compose trata-se de uma solução que provem diversos containers de aplicação.

Todo o comportamento do Docker Compose é gerenciado através do arquivo chamado docker-compose, semelhante ao Dockerfile, escrito em YAML (acrônimo recursivo para YAML Ain't Markup Language). Trata-se de um formato de codificação de dados legíveis por humanos, o que torna fácil de ler e entender o que um Compose faz.

Um exemplo prático de como funciona o Docker Compose é:

Imagine uma aplicação Java ou PHP e que essa aplicação depende de um banco de dados MySQL e, para disponibilizar essa aplicação na internet, será necessária a utilização de um proxy na frente.

Estes cenários replicado para Desenvolvimento, Homologação e Produção seriam muito mais trabalhosos e complicados de se implementar e gerenciar.

Outro ponto interessante, é que se esses ambientes estão todos em cloud e existe uma Instância/VM para cada aplicação, isso pode gerar custo, desperdício de recursos e tempo desnecessário gasto, sem contar o trabalho para montar toda a infra, em todos os momentos que surge um projeto novo.

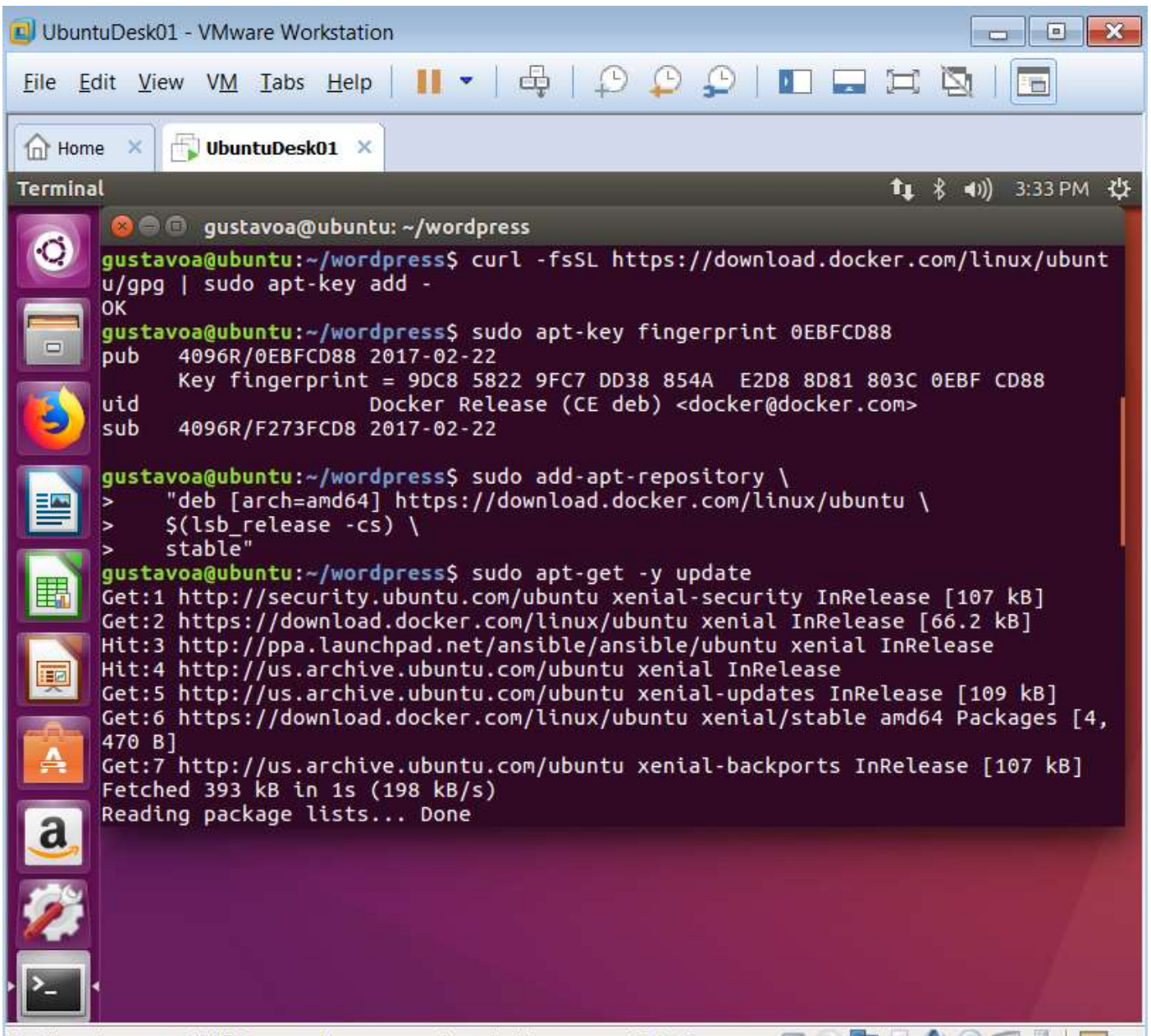
Por último, é que todos os processos supracitados ocorrem manualmente, causando um enorme esforço de recursos disponibilizados para isso.

Utilizando o Docker Compose é possível não só definir uma estrutura para atendimento as demandas citadas no exemplo acima, como também, estabelecer qual será ação que será tomada, caso um container venha a falhar. Se, por exemplo, o container de um banco de dados falhar, o arquivo Compose poderá ter uma instrução de subir outro imediatamente.

Ao longo deste documento a implementação de containers utilizando o Docker Compose será detalhada.

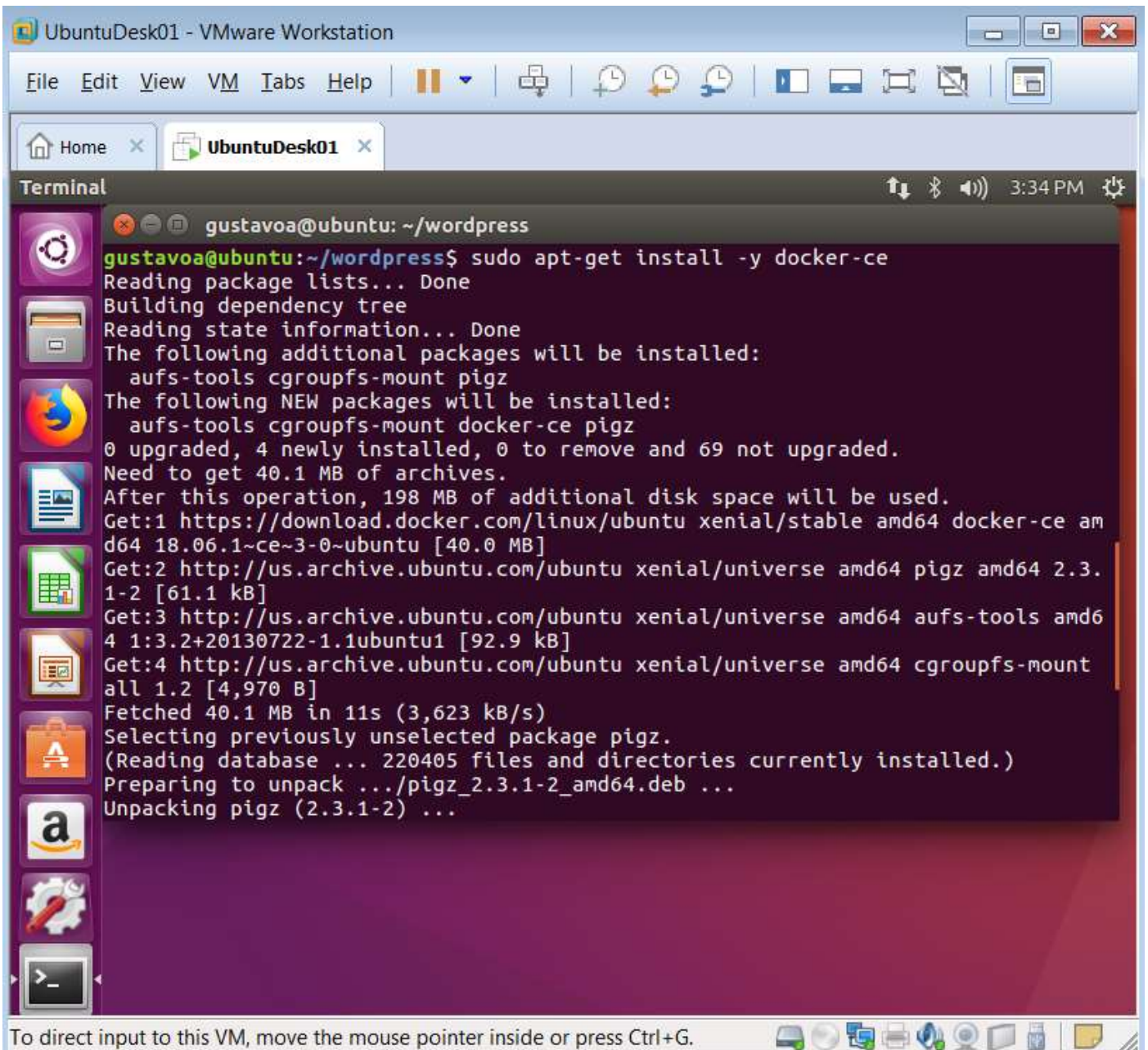
## **2- Documentação da aplicação:**

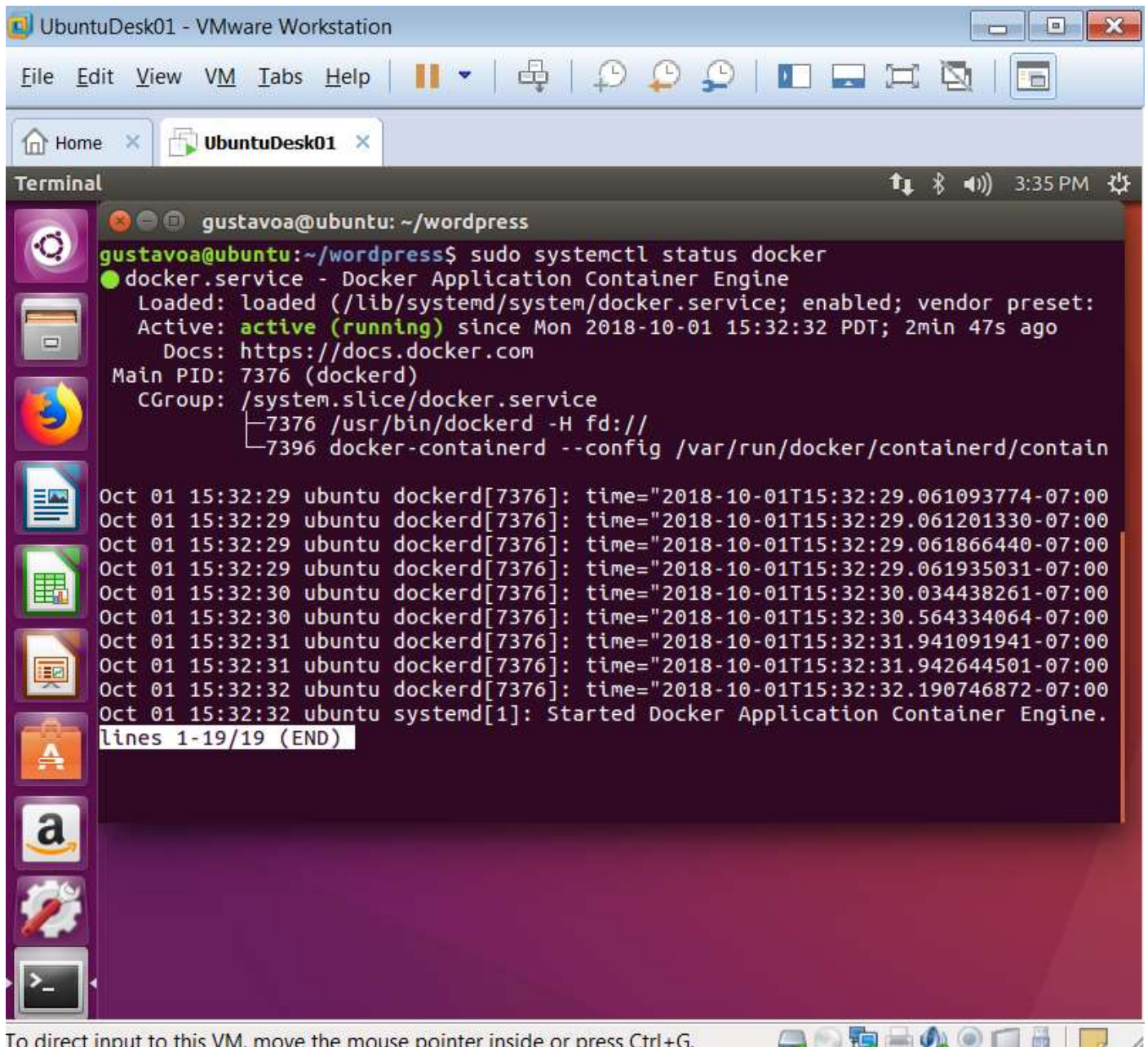
### **Detalhamento da instalação do Docker Compose:**



```
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
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
```





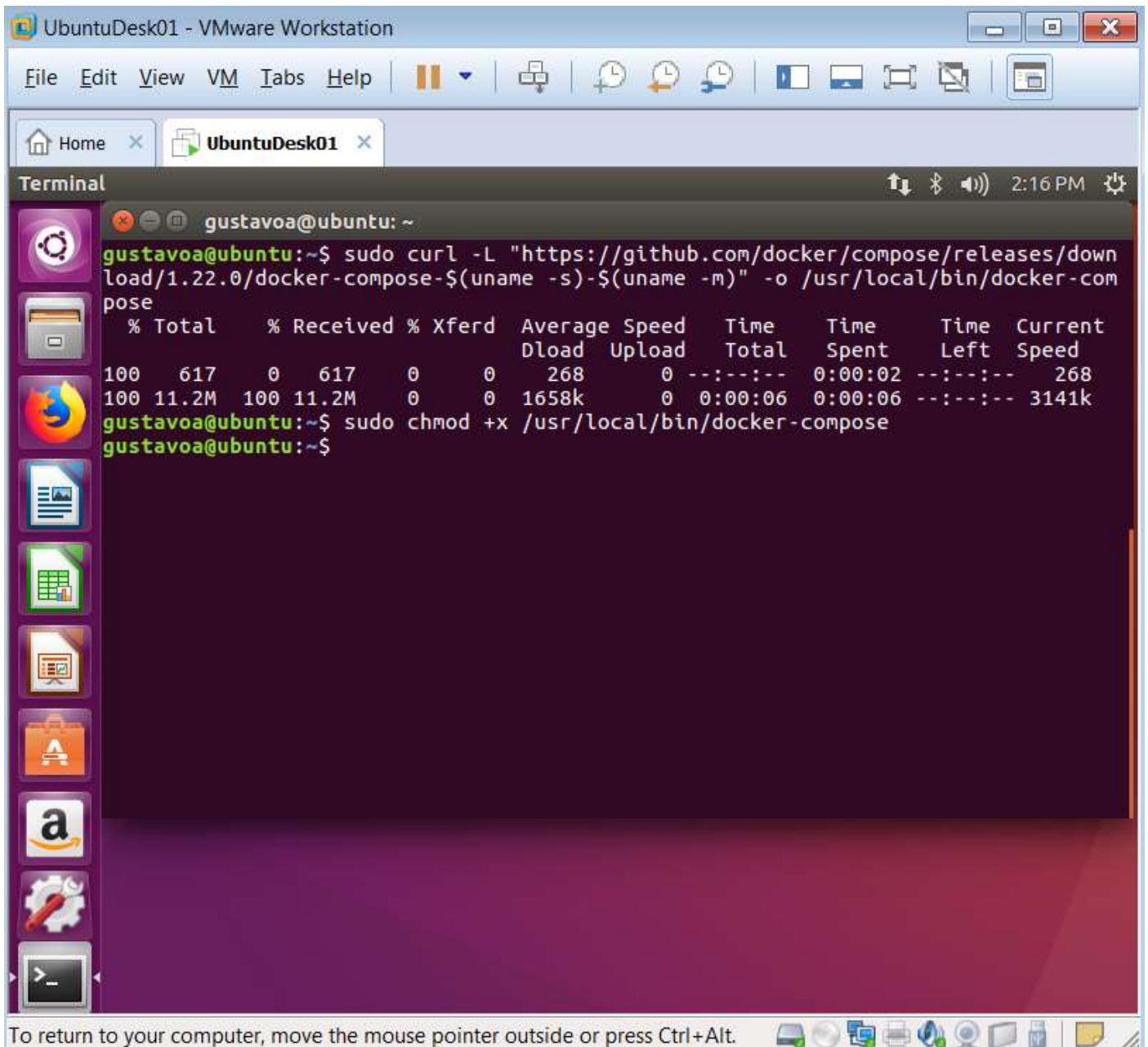


```
gustavoa@ubuntu: ~/wordpress
gustavoa@ubuntu:~/wordpress$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Mon 2018-10-01 15:32:32 PDT; 2min 47s ago
     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.





```

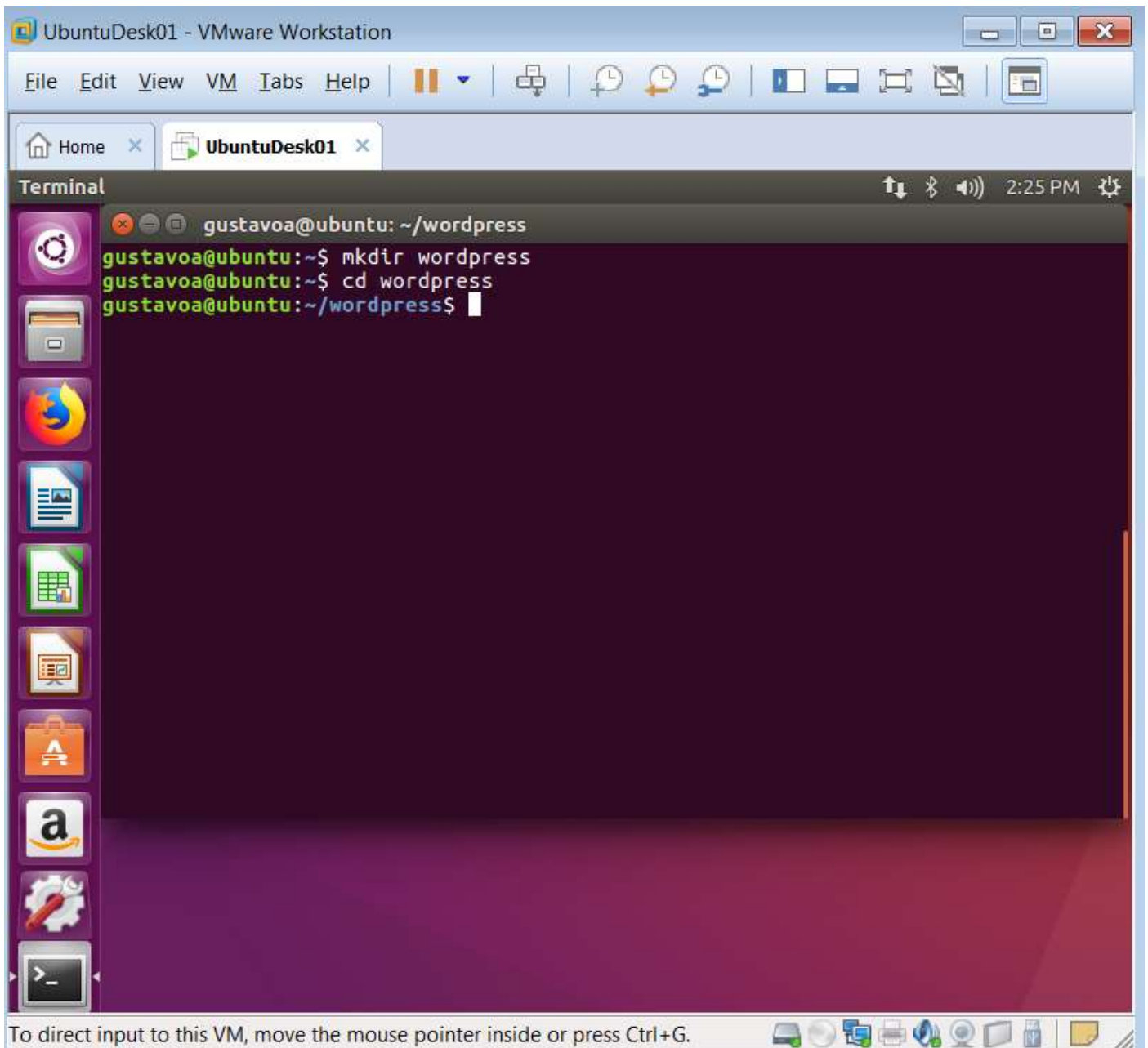
gustavoa@ubuntu: ~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.22.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
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:~$

```

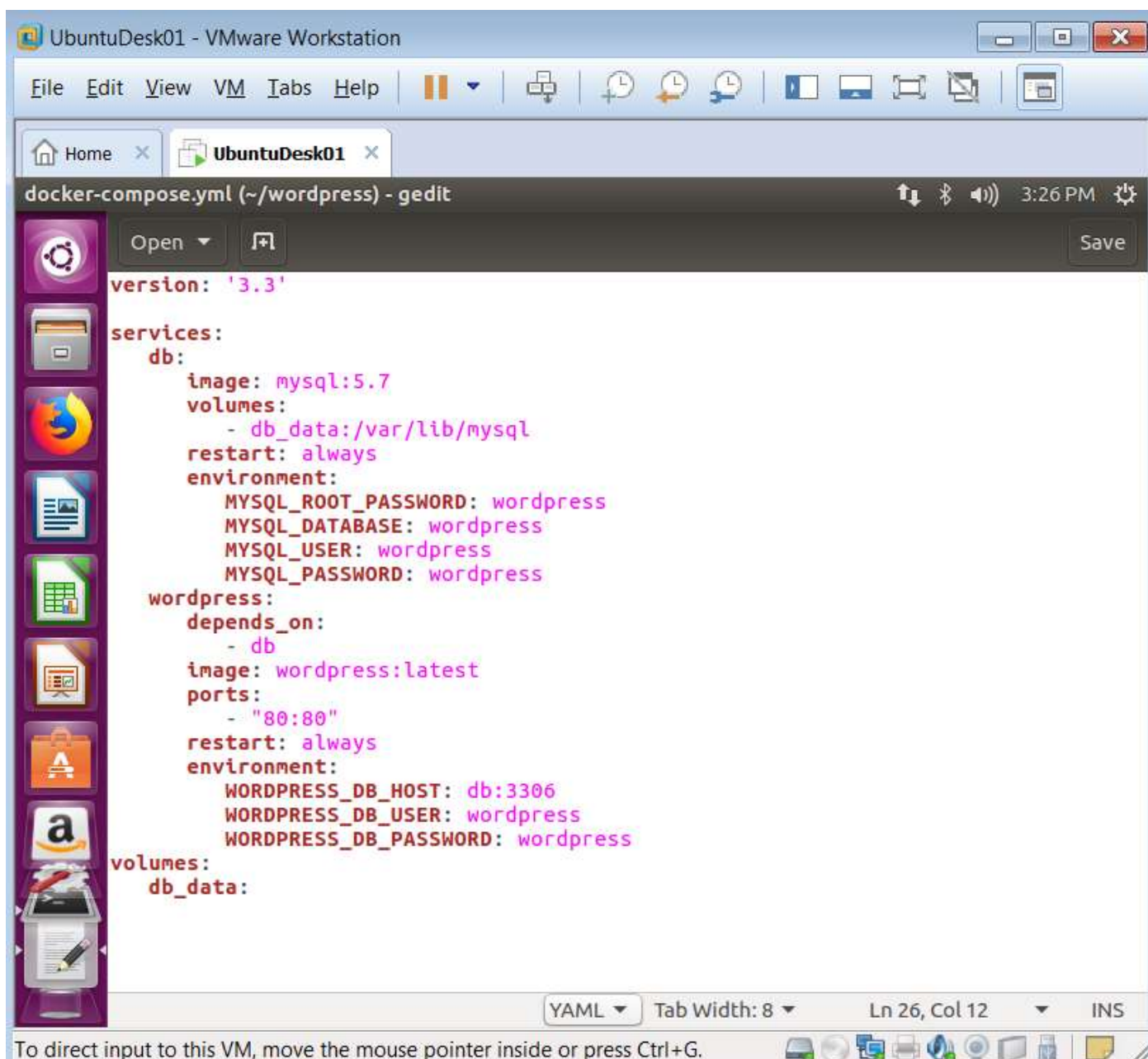
### 3- Comandos necessários para a execução dos containers:

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)**

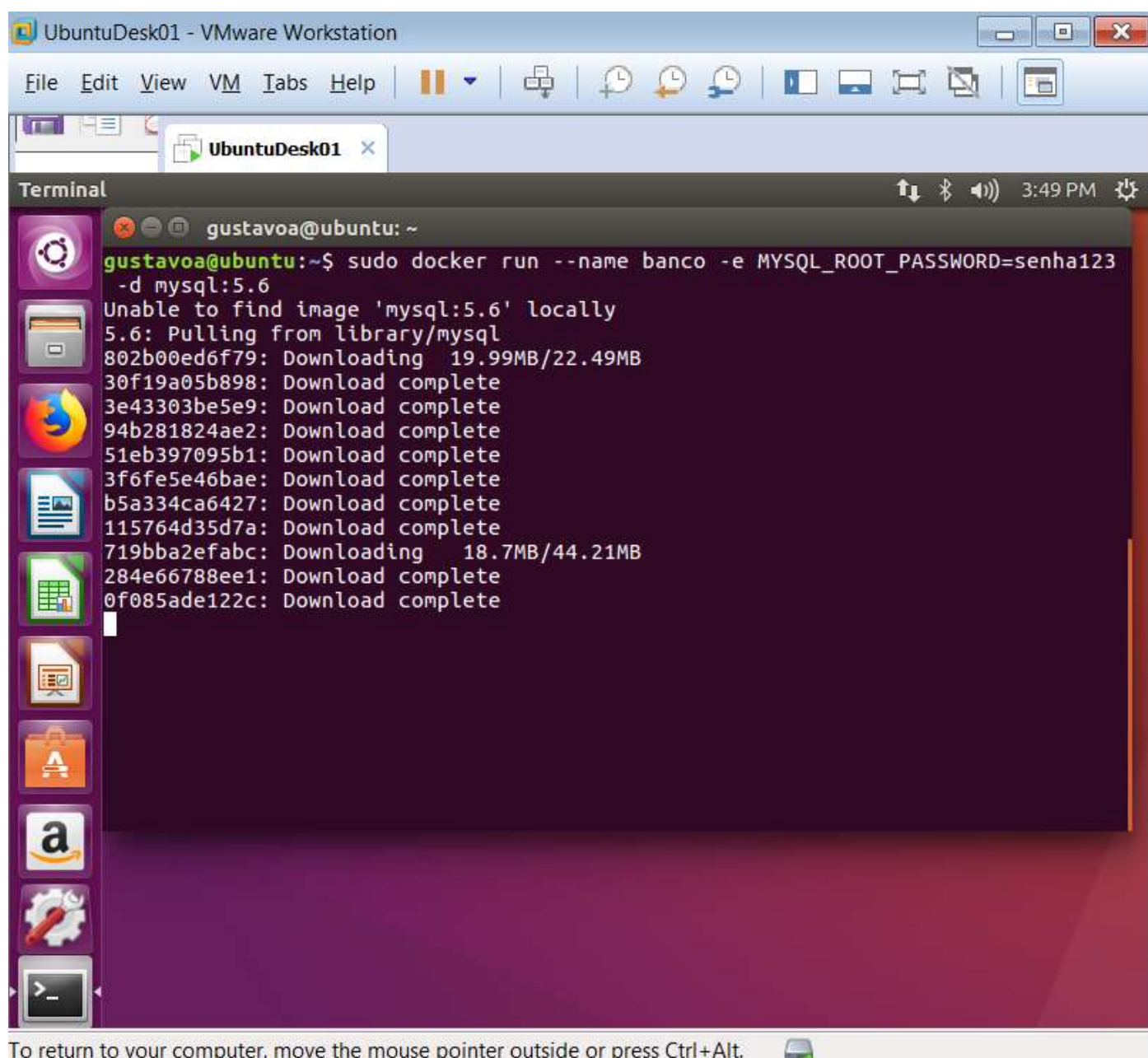


```
version: '3.3'

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:
```

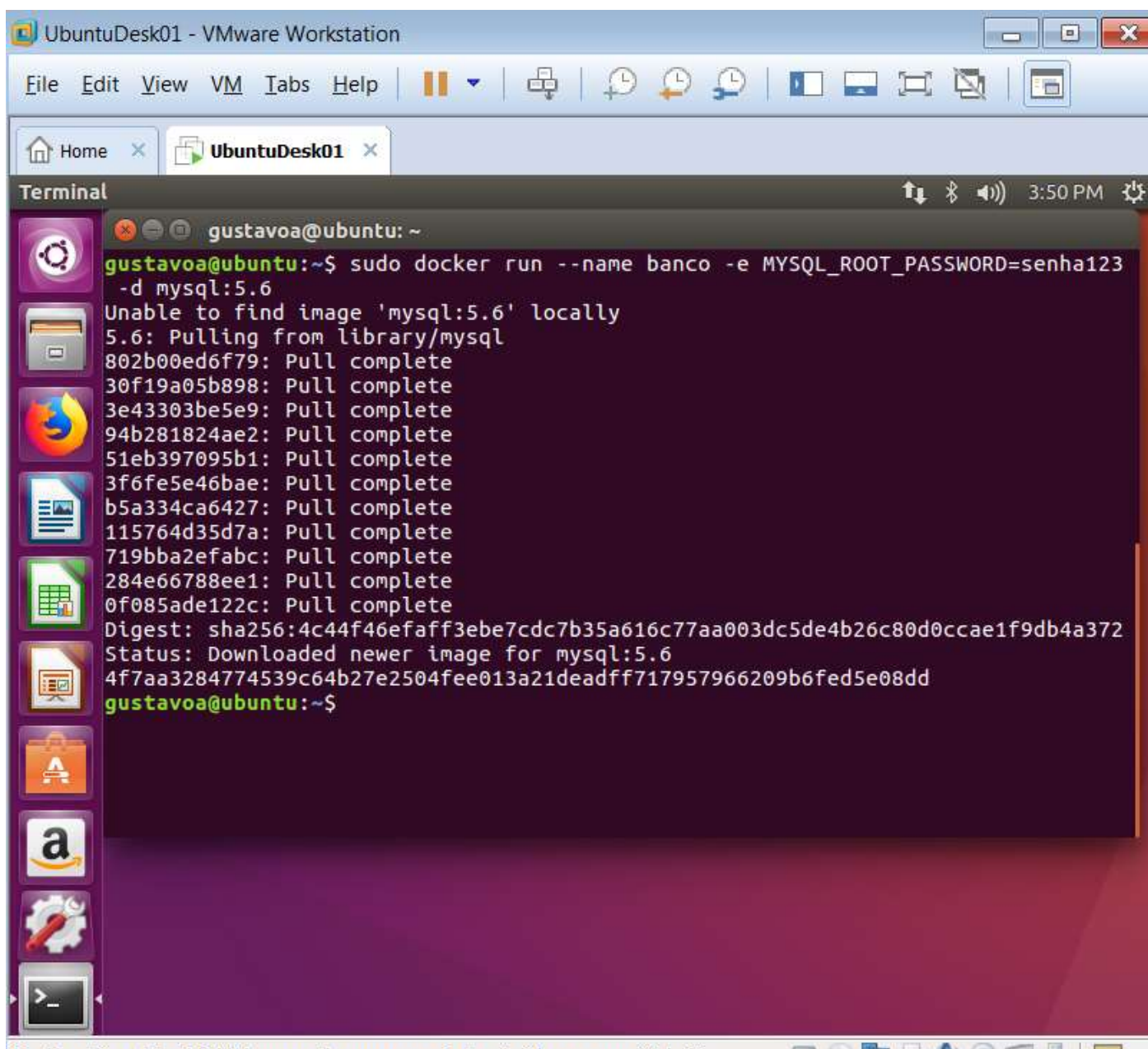
### Ativação dos containers (Banco de dados MySQL e Wordpress)

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
```

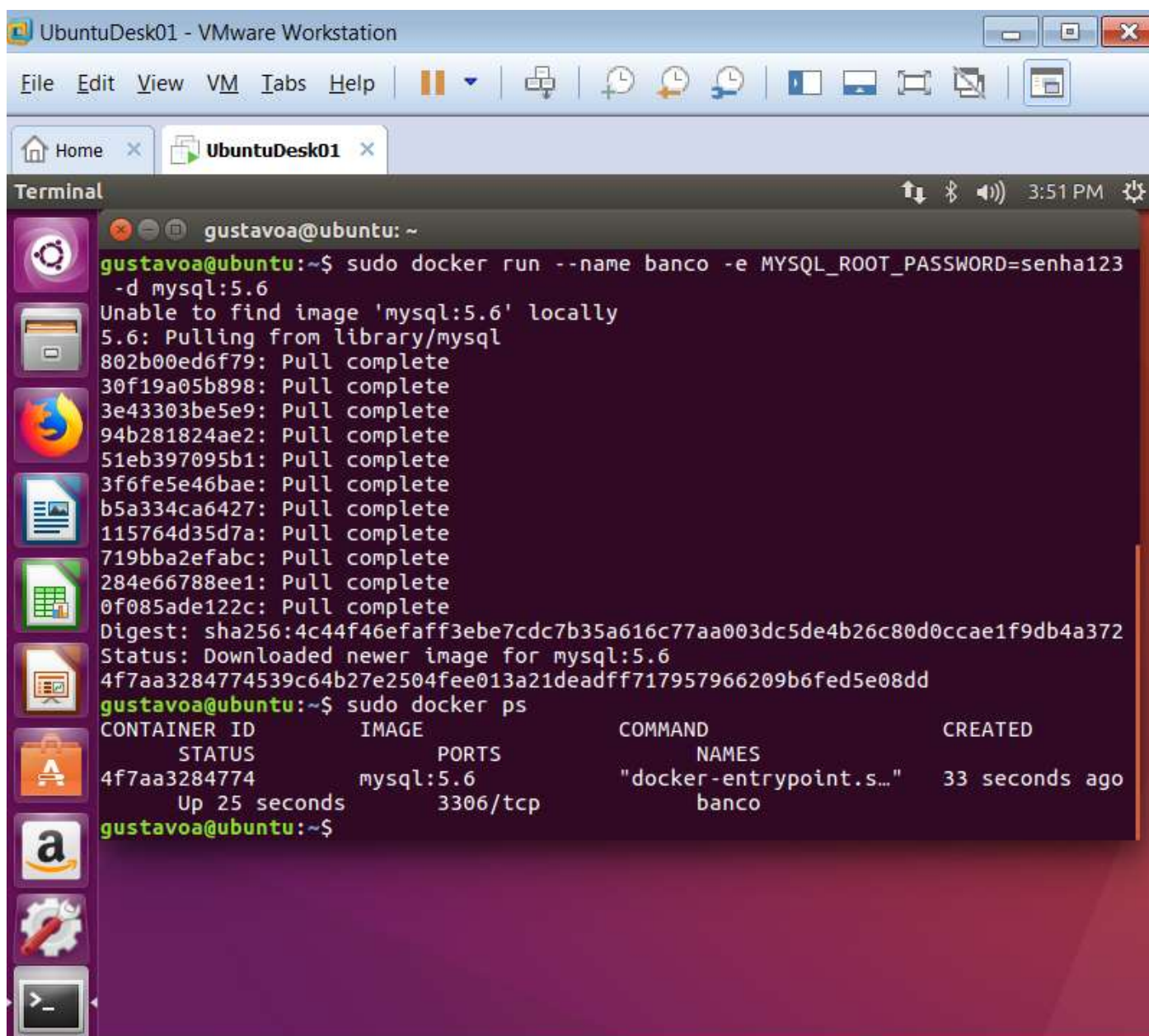
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.



```
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: ~$
```

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

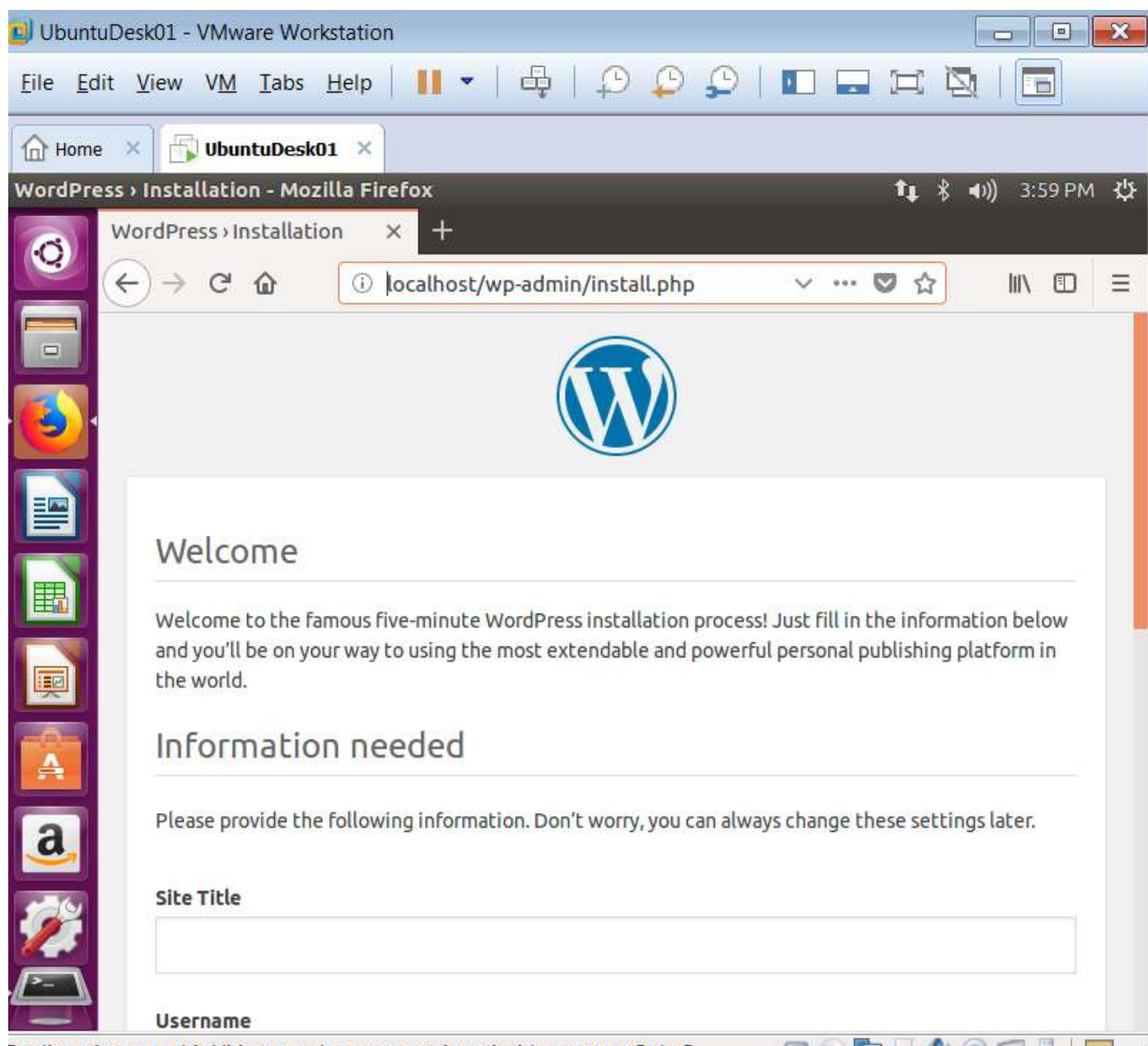




```
gustavoa@ubuntu:~$ sudo docker run --name banco -e MYSQL_ROOT_PASSWORD=senha123
-d mysql:5.6
Unable to find image 'mysql:5.6' locally
5.6: Pulling from library/mysql
802b00ed6f79: Pull complete
30f19a05b898: Pull complete
3e43303be5e9: Pull complete
94b281824ae2: Pull complete
51eb397095b1: Pull 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               COMMAND             CREATED
STATUS            PORTS              NAMES
4f7aa3284774        mysql:5.6          "docker-entrypoint.s..."   33 seconds ago
Up 25 seconds      3306/tcp           banco
gustavoa@ubuntu:~$
```

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

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



## 5- Playbook Ansible:

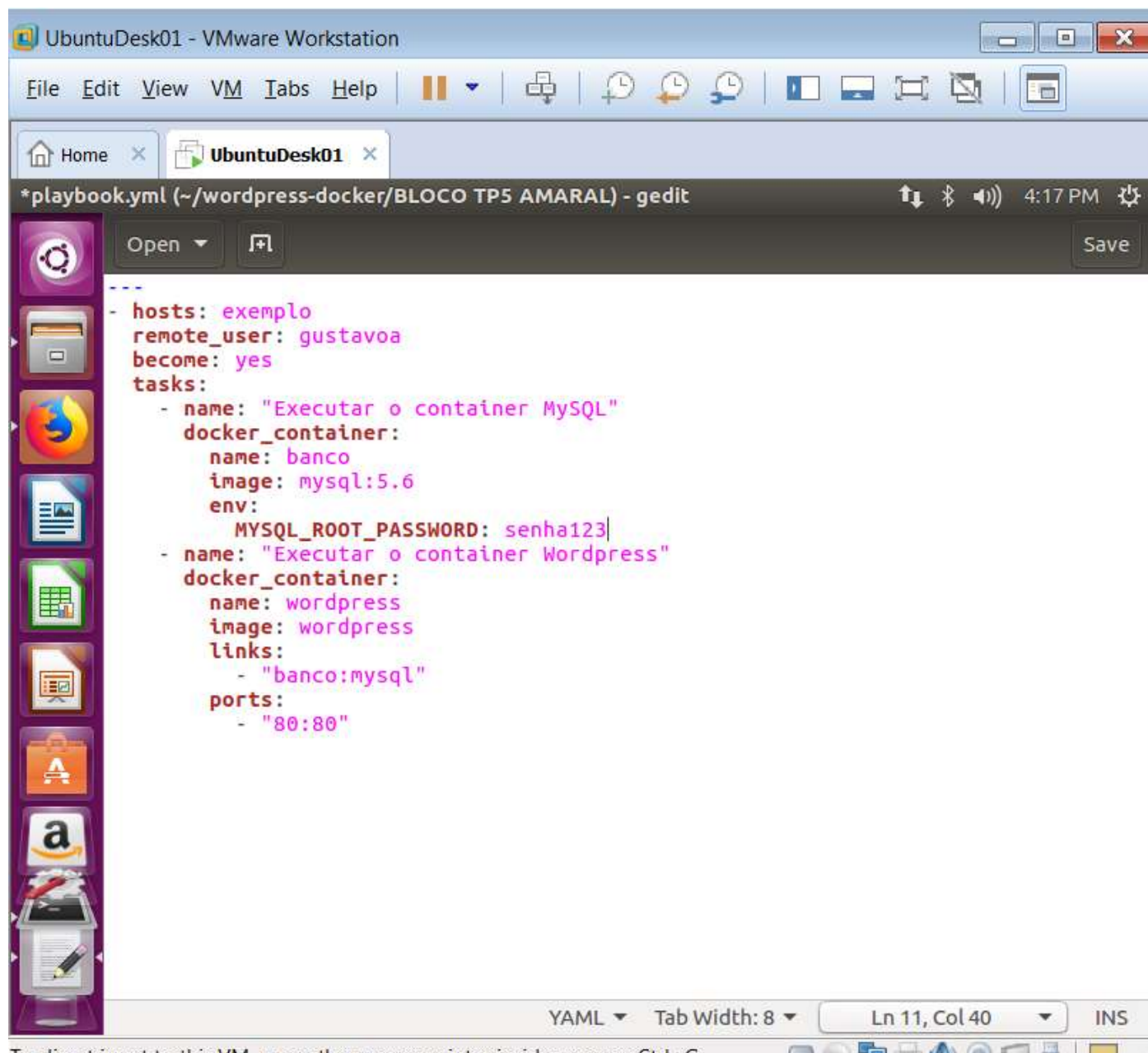
---

- 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"

## 6- Demonstração do playbook ansible funcional:



The screenshot shows a VMware Workstation window titled 'UbuntuDesk01 - VMware Workstation'. The main window displays a gedit editor with a file named '\*playbook.yml (~/.wordpress-docker/BLOCO TP5 AMARAL)'. The editor contains the following Ansible playbook content:

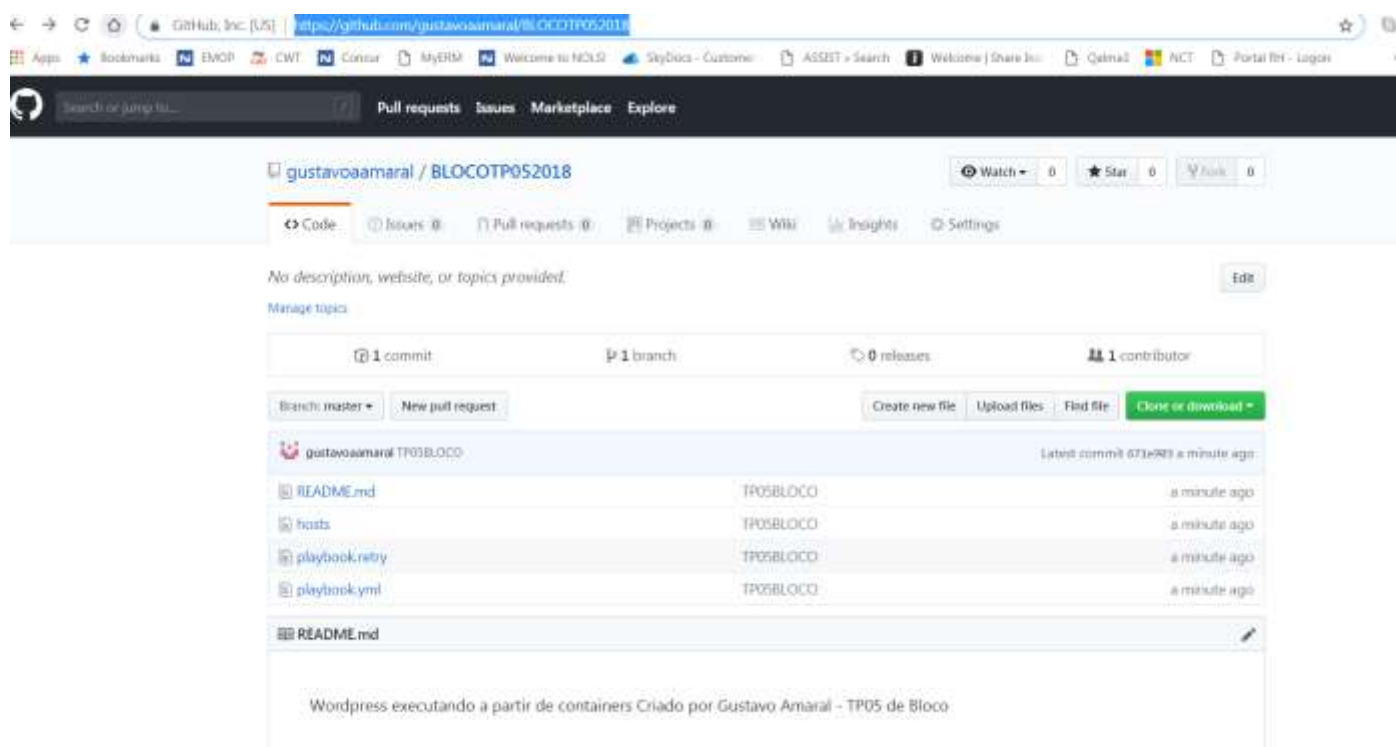
```
---
- hosts: exemplo
  remote_user: gustavo
  become: yes
  tasks:
    - name: "Executar o container MySQL"
      docker_container:
        name: banco
        image: mysql:5.6
        env:
          MYSQL_ROOT_PASSWORD: senha123
    - name: "Executar o container Wordpress"
      docker_container: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

The status bar at the bottom of the editor shows 'YAML', 'Tab Width: 8', 'Ln 11, Col 40', and 'INS'.

## 7- Playbook Ansible publicado no repositório Github:

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





## 8- 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.

## TP05 – Rubrica (Competência):

4. Levantar requisitos para um projeto de infraestrutura para aplicações  
O aluno selecionou uma aplicação com as características solicitadas?

4. Levantar requisitos para um projeto de infraestrutura para aplicações  
O aluno documentou a aplicação conforme requisitado?

4. Levantar requisitos para um projeto de infraestrutura para aplicações  
O aluno apresentou os comandos necessários para a execução dos containers?

4. Levantar requisitos para um projeto de infraestrutura para aplicações  
O aluno conseguiu demonstrar que os containers estavam acessíveis por um cliente, usando capturas de tela?

4. Levantar requisitos para um projeto de infraestrutura para aplicações  
O aluno apresentou um playbook Ansible válido, ainda que a aplicação não seja executada corretamente?

TP5

4. Levantar requisitos para um projeto de infraestrutura para aplicações  
O aluno apresentou um playbook capaz de executar a aplicação pretendida com sucesso?

TP5

4. Levantar requisitos para um projeto de infraestrutura para aplicações

O aluno publicou o seu playbook em um repositório GitHub e forneceu a URL para verificação?

TP5

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?

## TP05 – Referências :

Acesso em: 22/09/2018

- <https://lms.infnet.edu.br/moodle/mod/page/view.php?id=94736>
- <https://lms.infnet.edu.br/moodle/mod/assign/view.php?id=94738>
- <https://lms.infnet.edu.br/moodle/mod/assign/view.php?id=94746>
- <https://www.youtube.com/watch?v=o3Cz1jyEahE>
- <http://helton.de/Blog/article/pt/267>
- <https://lms.infnet.edu.br/moodle/mod/page/view.php?id=94744>
- <https://www.ansible.com/resources/videos/quick-start-video>
- <https://github.com/gustavoamaral/ProjblocoTP03>
- <https://lms.infnet.edu.br/moodle/mod/page/view.php?id=94751>
- <https://imasters.com.br/banco-de-dados/docker-compose-o-que-e-para-que-serve-o-que-serve>
- <http://blog.alura.com.br/compondo-uma-aplicacao-com-o-docker-compose/>
- <https://www.mundodocker.com.br/tag/dica-docker/>
- <https://www.concrete.com.br/2016/09/16/aws-e-as-aplicacoes-dockerizadas/>
- <https://lms.infnet.edu.br/moodle/mod/page/view.php?id=94759>
- <https://store.docker.com/images/c14a56d6-07e4-464b-b71c-4b24dc7f1836?tab=description>

=