

**INSTITUTO INFNET**  
**ESCOLA SUPERIOR DE TECNOLOGIA DA INFORMAÇÃO**  
**GRADUAÇÃO EM GESTÃO DE TI**



**PROJETO DE BLOCO**  
**ARQUITETURA DE INFRAESTRUTURA DE APLICAÇÕES**  
**TESTE DE PERFORMANCE – TP6**  
**ALUNO: Higor Martins Nery**  
**E-MAIL: higor.nery@al.infnet.edu.br**  
**TURMA: GGTI - NOITE**

## SUMÁRIO

1	INTRODUÇÃO .....	1
2	OBJETIVO .....	1
3	JUSTIFICATIVA .....	1
4	DESCRIÇÃO DA TAREFA.....	1
5	INFORMAÇÕES DO PROJETO – TESTE DE PERFORMANCE 1.....	2
5.1	DETALHES TÉCNICOS E PRÉ-REQUISITOS .....	2
6	PLANO DE IMPLEMENTAÇÃO.....	3
7	IMPLEMENTAÇÃO – TESTES DE PERFORMANCE 2 E 3.....	4
7.1	Execução do playbook.....	4
7.2	Configurações do GitHub e estrutura de diretórios.....	6
8	APLICAÇÃO SELECIONADA – TESTES DE PERFORMANCE 4 E 5 .....	8
8.1	Implementação do Docker GLPI.....	10
8.2	Ansible Docker GLPI.....	13
9	PROPOSTA DE SERVIÇOS - TESTE DE PERFORMANCE 6 .....	14
9.1	OBJETIVO DA PROPOSTA.....	15
9.2	RESUMO DA PROPOSTA.....	15
9.3	DESCRIÇÃO DO CENÁRIO .....	15
9.4	SOLUÇÃO PROPOSTA.....	16
9.5	REQUISITOS MÍNIMOS.....	16
9.6	INFRAESTRUTURA PARA APLICAÇÃO .....	18
10	DESCRIÇÃO DA TECNOLOGIA UTILIZADA – TP 7.....	19
11	COMPARATIVO – TESTE DE PERFORMANCE 7 .....	19
12	PLANO DE IMPLEMENTAÇÃO – TP7 E TP8 .....	20
12.1	Implementação do EC2 – TP7 e TP8.....	20
12.2	Configuração do Ansible Docker – TP7 e TP8 .....	22
13	APLICAÇÃO EM FUNCIONAMENTO – TP8.....	23
14	CRONOGRAMA DO PROJETO – TP 7 .....	24
15	CONSIDERAÇÕES SOBRE O PROJETO - TP9.....	24
15.1	Melhorias sugeridas – TP9 .....	25
15.2	Repositório para o projeto .....	25

## **1 INTRODUÇÃO**

Como forma de avaliação o Instituto Infnet, desenvolveu o teste de performance para a avaliação das disciplinas regulares e de projeto de bloco. Esta avaliação consiste em aplicar o conteúdo lecionado nas disciplinas regulares e efetuar uma junção dos conhecimentos para execução baseado em um estudo de caso.

## **2 OBJETIVO**

O teste de performance em questão tem como objetivo aplicar as habilidades desenvolvidas durante as disciplinas regulares do semestre em um projeto de baseado em um estudo de caso.

## **3 JUSTIFICATIVA**

O mercado de trabalho necessita de profissionais qualificados para exercerem funções técnicas e principalmente gerenciais na área de tecnologia da informação. É pré-requisito para qualquer função, tanto técnica quanto gerencial que o colaborador saiba analisar e sugerir melhorias em qualquer cenário.

## **4 DESCRIÇÃO DA TAREFA**

Durante o semestre vigente, será desenvolvida a implementação de uma aplicação virtualizada, bem como sua infraestrutura e controle de versões. Por fim, o objetivo principal é que tal aplicação seja de total utilidade e que possa agregar valor de TI ao negócio.

## **5      INFORMAÇÕES DO PROJETO – TESTE DE PERFORMANCE 1**

Visando conceder maior controle da área de tecnologia da informação para a alta direção o projeto em questão será implementado a aplicação GLPI. Tal ferramenta fará com que os a TI seja transformada em “números” trazendo alto controle com chamados, inventário de estações de trabalho, servidores e dispositivos móveis.

### **5.1      DETALHES TÉCNICOS E PRÉ-REQUISITOS**

Visando melhor entendimento quanto a disponibilização de uma aplicação, devemos minimamente apontar os detalhes técnicos para análise de tal procedimento, o qual segue demonstrado:

Características do servidor web necessárias para executar GLPI.

Servidor web

GLPI requer um servidor web que suporta PHP, tais como:

Apache 2 ou superior (<http://httpd.apache.org>)

Microsoft IIS (<http://www.iis.net>).

PHP

GLPI requer PHP versão 5.3 ou superior (<http://www.php.net>).

Extensões PHP necessárias

As seguintes extensões PHP são necessárias para o bom funcionamento do aplicativo:

JSON: suporte de formato dados estruturados JSON

Mbstring: gerenciamento de cadeia de caracteres de vários bytes

MySQL: conexão com o banco de dados

Sessão: Usuários de suporte de sessão

Extensões PHP opcionais, mas recomendadas

As seguintes extensões são necessárias para recursos opcionais do aplicativo:

CLI: Usando PHP a partir da linha de comando para ações automáticas

CURL: CAS para autenticação

DOMXML: CAS para autenticação

GD: geração de imagens

IMAP: servidores de correio usam para coletar os ingressos ou a autenticação do usuário

LDAP: uso de um diretório externo para autenticação

OpenSSL: comunicação criptografada

Tais detalhes técnicos proporcionarão uma ambiente web totalmente disponível e de fácil interação com o usuário.

Todo controle de versões é efetuado via GIT HUB, nos fazendo com que tenhamos maior controle sobre as mudanças.

## **6 PLANO DE IMPLEMENTAÇÃO**

Para que haja maior segurança durante o período de implementação, faremos o processo em três etapas: teste, homologação e produção.

- Testes – A aplicação será instalada em um servidor, visando a criação de regras aplicadas para o negócio e testaremos novas integrações do sistema
- Homologação - Todos os testes bem-sucedidos serão enviados para a etapa de homologação para que os usuários possam se familiarizar com as novas funcionalidades
- Produção - Por fim, as novas funcionalidades serão aplicadas na produção para que os clientes possam usar no seu dia a dia.

## 7 IMPLEMENTAÇÃO – TESTES DE PERFORMANCE 2 E 3

Como parte do processo de implementação do GLPI, via ansible será efetuado a configuração de um playbook, juntamente com sua estrutura de diretórios e a estrutura montada no github.

### 7.1 Execução do playbook

---

- hosts: GLPI

remote\_user: higor

become: yes

tasks:

- name: Instala o Serviço NTP

apt: name=ntp state=present

- name: Garante que o NTP está rodando

service: name=ntp state=started enabled=yes

playbook.yml - GLPI - Visual Studio Code

EXPLORER

- OPEN EDITORS
  - Welcome
  - ! playbook.yml M
  - hosts
- GLPI
  - defaults
  - files
  - handlers
  - meta
  - roles
  - tasks
  - templates
  - tests
  - vars
  - hosts
- ! playbook.yml M
- README.md

```
1  ---
2  - hosts: GLPI
3    remote_user: higor
4    become: yes
5    tasks:
6      - name: Instala o Serviço NTP
7        apt: name=ntp state=present
8      - name: Garante que o NTP está rodando
9        service: name=ntp state=started enabled=yes
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

```
higor@GLPI:~/ansible-codes/GLPI$ ansible-playbook playbook.yml
[WARNING]: Could not match supplied host pattern, ignoring: all

[WARNING]: provided hosts list is empty, only localhost is available

[WARNING]: Could not match supplied host pattern, ignoring: GLPI

PLAY [GLPI] *****
**
skipping: no hosts matched

PLAY RECAP *****
**

higor@GLPI:~/ansible-codes/GLPI$
```

Ln 9, Col 48 Spaces: 2 UTF-8 LF YAML

## 7.2 Configurações do GitHub e estrutura de diretórios

Seguem abaixo configurações efetuadas no github:

- Link do profile do GitHub


 [higornery](#) / [Ansible-GLPI](#)

**Quick setup — if you've done this kind of thing before**

or


HTTPS

SSH




We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**



```
echo "# Ansible-GLPI" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:higornery/Ansible-GLPI.git
git push -u origin master
```

**...or push an existing repository from the command line**



```
git remote add origin git@github.com:higornery/Ansible-GLPI.git
git push -u origin master
```

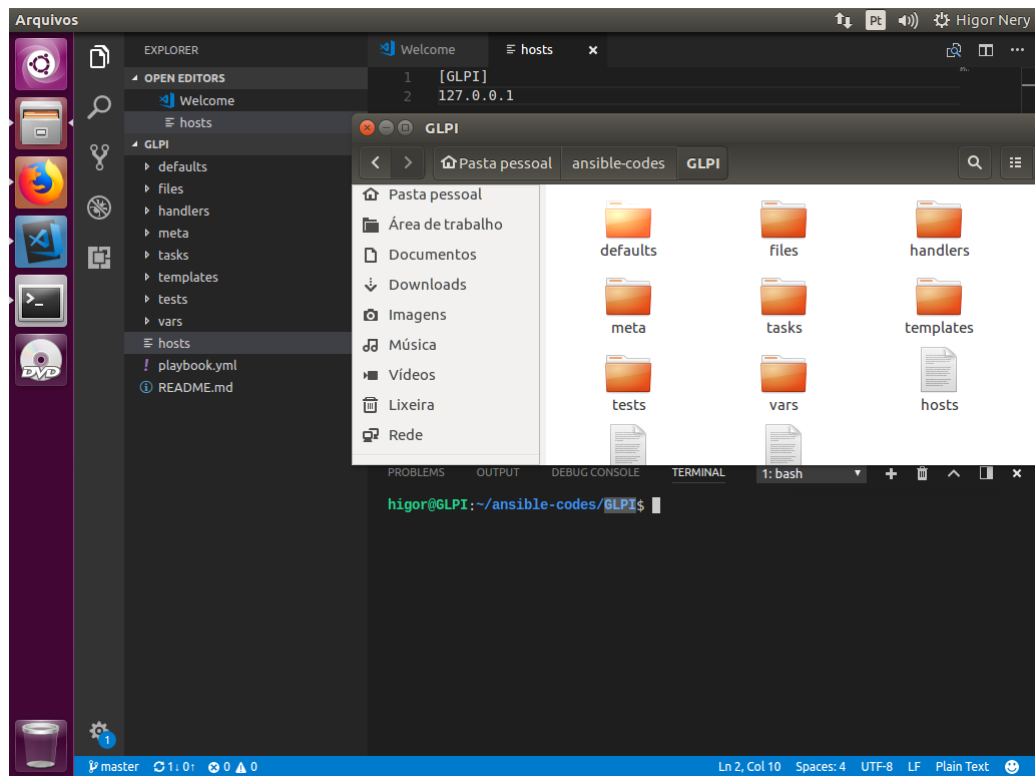
**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

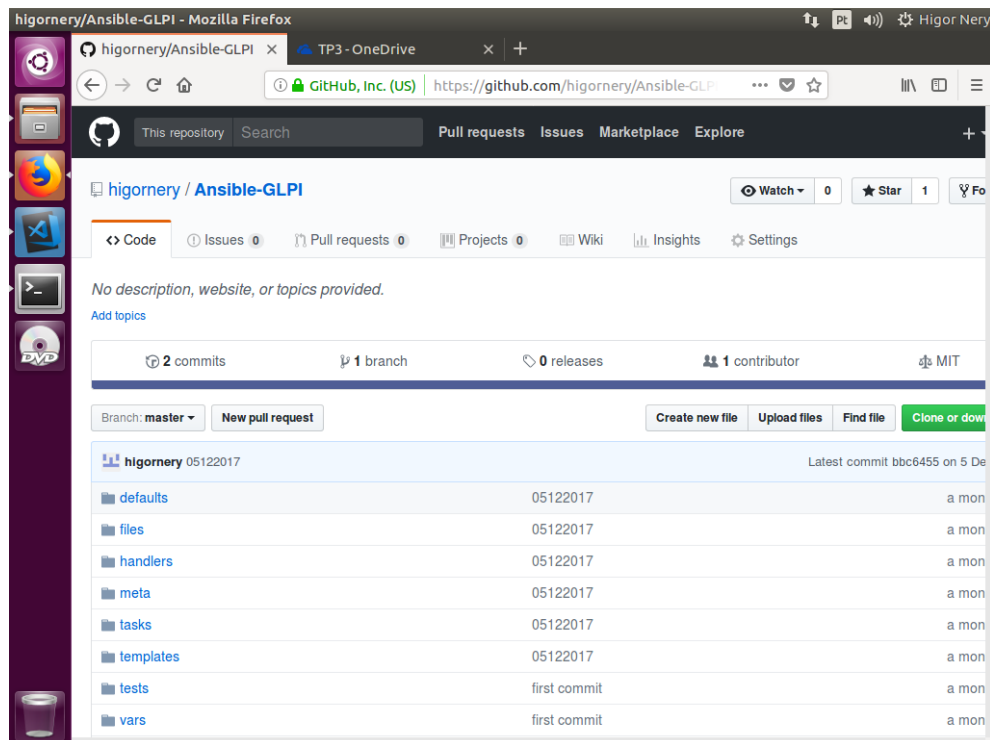
 **ProTip!** Use the URL for this page when adding GitHub as a remote.



- Estrutura de diretórios montada para o ansible



- Push no git hub



## **8 APLICAÇÃO SELECIONADA – TESTES DE PERFORMANCE 4 E 5**

Para que os objetivos relacionados a área de TI da empresa em questão sejam alcançados, faremos a implementação do GLPI e tal atividade tem a rela necessidade de fazermos a implementação dos seus pré-requisitos.

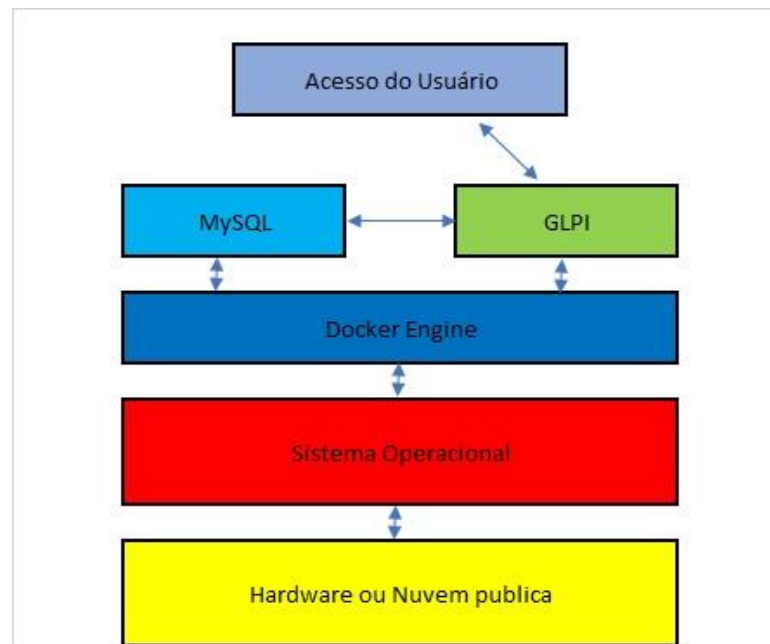
Os pré-requisitos para que aplicação tenha um bom funcionamento são os seguintes:

- Apache + PHP – Tal pré-requisito fara com que tenhamos a possibilidade de executar a aplicação web como um todo.
- MySQL – Este requisito será o SGBD (Sistema de gerenciamento de banco de dados) do sistema, ou seja, todos os dados gerados pela aplicação serão armazenados neste banco de dados.

Tanto a aplicação quanto o banco de dados serão configurados em servidores virtuais, que irão nos proporcionar maior escalabilidade do ambiente caso seja necessário. Para o ambiente solicitado, será efetuada a implementação da tecnologia de container, e mesma utilizada será o Docker

“O Docker é uma plataforma de virtualização Open Source escrita na linguagem Go (linguagem de alto desempenho criada pela Google) e mantida pela Docker Inc (empresa localizada na Califórnia – EUA), que permite a criação e administração de ambientes isolados hospedados em um sistema operacional (SO). A plataforma, permite que seja encapsulado uma aplicação ou, até mesmo um sistema operacional inteiro dentro desses ambientes, que são chamados de containers.” (Correa, 30)

A imagem abaixo mostra como ficará a arquitetura do GLPI virtualizado no Docker.



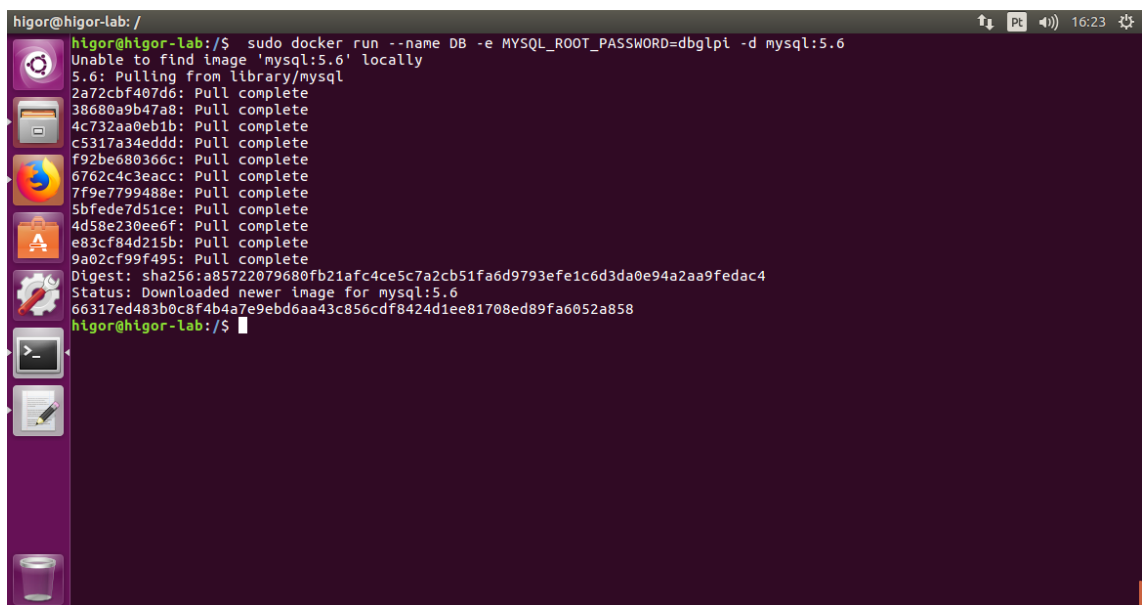
Para que possamos ter menor custo no ambiente, a implementação deverá ser apontada para uma nuvem pública.

## 8.1 Implementação do Docker GLPI

As imagens demonstram a implementação do GLPI no Docker. Vale ressaltar que o GLPI possui algumas dependências que serão instaladas em containers separados. As dependências o Mysql e o Apache, juntamente com o PHP.

Inicialmente a será efetuada a configuração do container do MySql, conforme imagem abaixo:

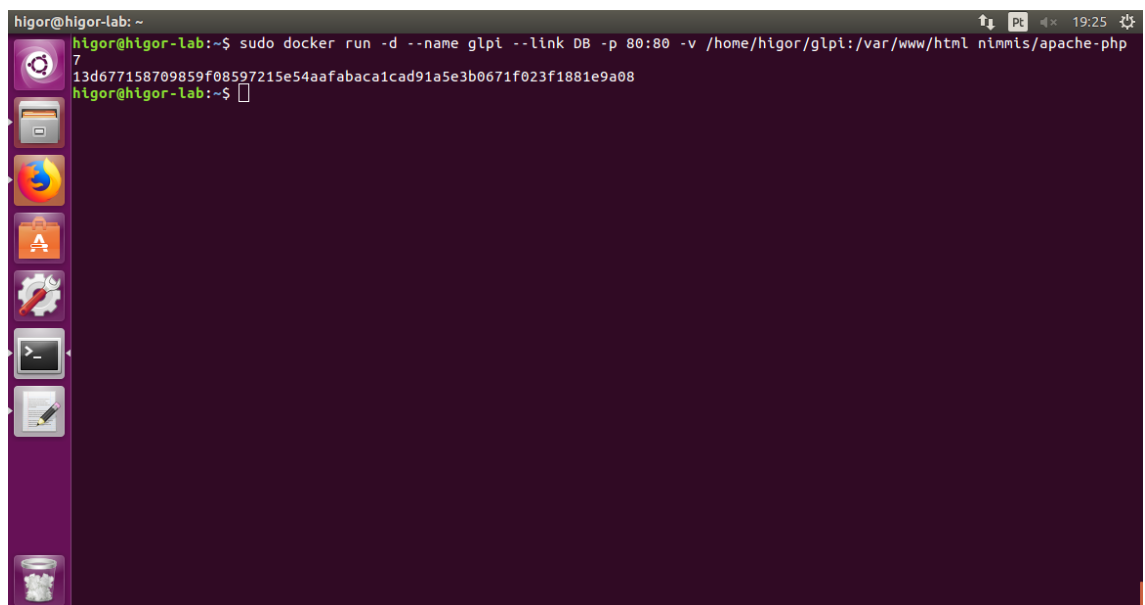
- O comando utilizado para a configuração do container foi o **sudo docker run --name DB -e MYSQL\_ROOT\_PASSWORD=dbglpi -d mysql:5.6**

A terminal window titled 'higor@higor-lab: /' showing the execution of the command 'sudo docker run --name DB -e MYSQL\_ROOT\_PASSWORD=dbglpi -d mysql:5.6'. The output shows the Docker engine pulling the 'mysql:5.6' image from the library. It lists several layers being pulled, all marked as 'Pull complete'. The digest is 'sha256:a85722079680fb21afc4ce5c7a2cb51fa6d9793efe1c6d3da0e94a2aa9fedac4'. The status is 'Downloaded newer image for mysql:5.6'. The final output is the container ID '66317ed483b0c8f4b4a7e9ebd6aa43c856cdf8424d1ee81708ed89fa6052a858' followed by a prompt 'higor@higor-lab: /\$'.

```
higor@higor-lab: /$ sudo docker run --name DB -e MYSQL_ROOT_PASSWORD=dbglpi -d mysql:5.6
Unable to find image 'mysql:5.6' locally
5.6: Pulling from library/mysql
2a72cbf407d6: Pull complete
38680a9b47a8: Pull complete
4c732aa0eb1b: Pull complete
c5317a34eddd: Pull complete
f92be680366c: Pull complete
6762c4c3eacc: Pull complete
7f9e7799488e: Pull complete
5bfede7d51ce: Pull complete
4d58e230ee6f: Pull complete
e83cf84d215b: Pull complete
9a02cf99f495: Pull complete
Digest: sha256:a85722079680fb21afc4ce5c7a2cb51fa6d9793efe1c6d3da0e94a2aa9fedac4
Status: Downloaded newer image for mysql:5.6
66317ed483b0c8f4b4a7e9ebd6aa43c856cdf8424d1ee81708ed89fa6052a858
higor@higor-lab: /$
```

- Após a configuração do container do MYSQL, será efetuada a configuração do container do APACHE+PHP. Neste container foi efetuado o mapeamento do diretório físico para o diretório do container. Tal configuração trará maior facilidade para criação de rotinas de backup e versionamento da aplicação. O comando abaixo foi executado para obtenção do resultado:

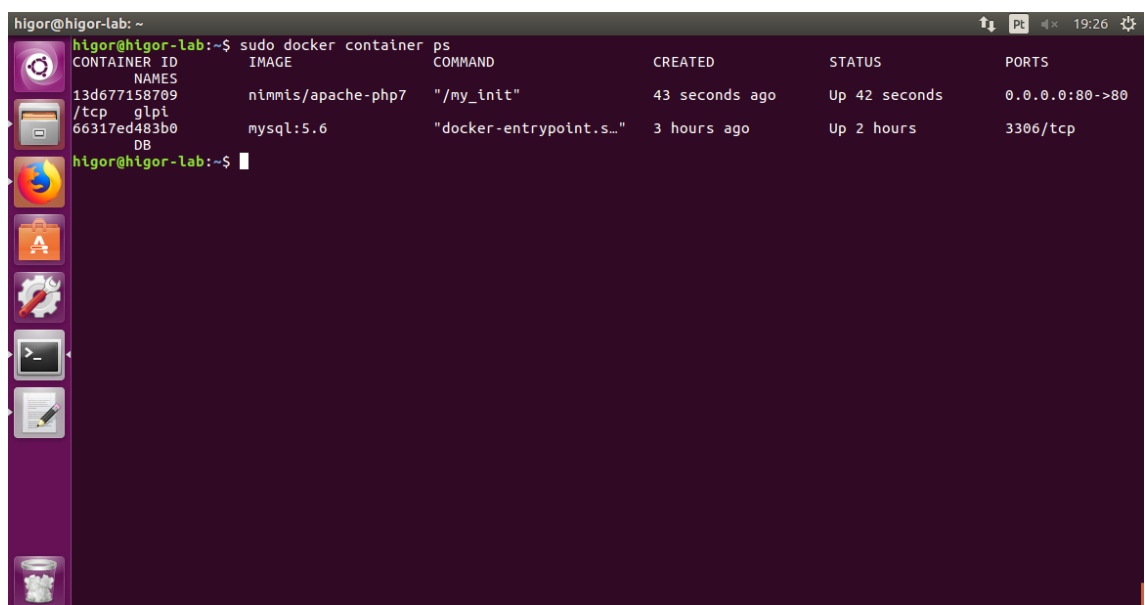
**sudo docker run -d --name glpi --link DB -p 80:80 -v /home/higor/glpi:/var/www/html nimmis/apache-php7**



```

higor@higor-lab: ~
higor@higor-lab:~$ sudo docker run -d --name glpi --link DB -p 80:80 -v /home/higor/glpi:/var/www/html nimmis/apache-php7
13d677158709859f08597215e54aafabaca1cad91a5e3b0671f023f1881e9a08
higor@higor-lab:~$
  
```

- **Containers em execução**



```

higor@higor-lab: ~
higor@higor-lab:~$ sudo docker container ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
13d677158709   nimmis/apache-"/my_init"         43 seconds ago Up 42 seconds 0.0.0.0:80->80
66317ed483b0   mysql:5.6     "docker-entrypoint.s..." 3 hours ago   Up 2 hours    3306/tcp
higor@higor-lab:~$
  
```

- Aplicação em funcionamento. Vale ressaltar que a mesma foi acessada fora da máquina virtual.

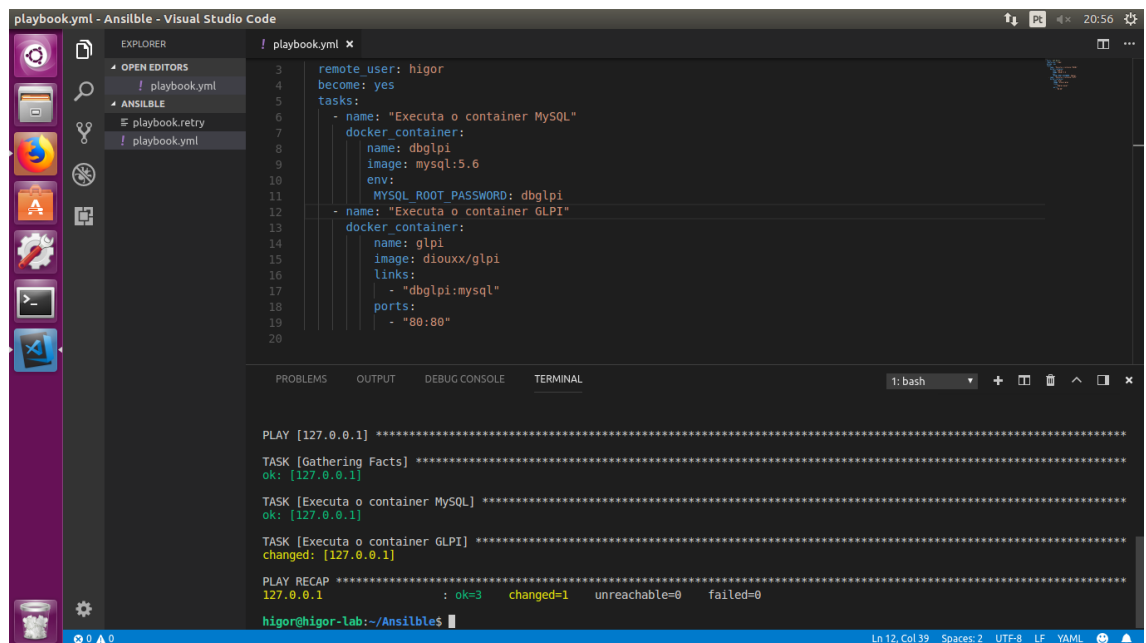


## 8.2 Ansible Docker GLPI

Buscando maior eficiência no processo de implementação da aplicação será efetuada implementação no Ansible, ferramenta de automatização de aplicações. Os códigos gerados terão o controle de versão efetuados via Git hub.

Pelo git, faremos a implementação de dois containers, um para o mysql e outro container que terá o apache, php e a aplicação do GLPI.

- A imagem abaixo mostra a execução do playbook ansible



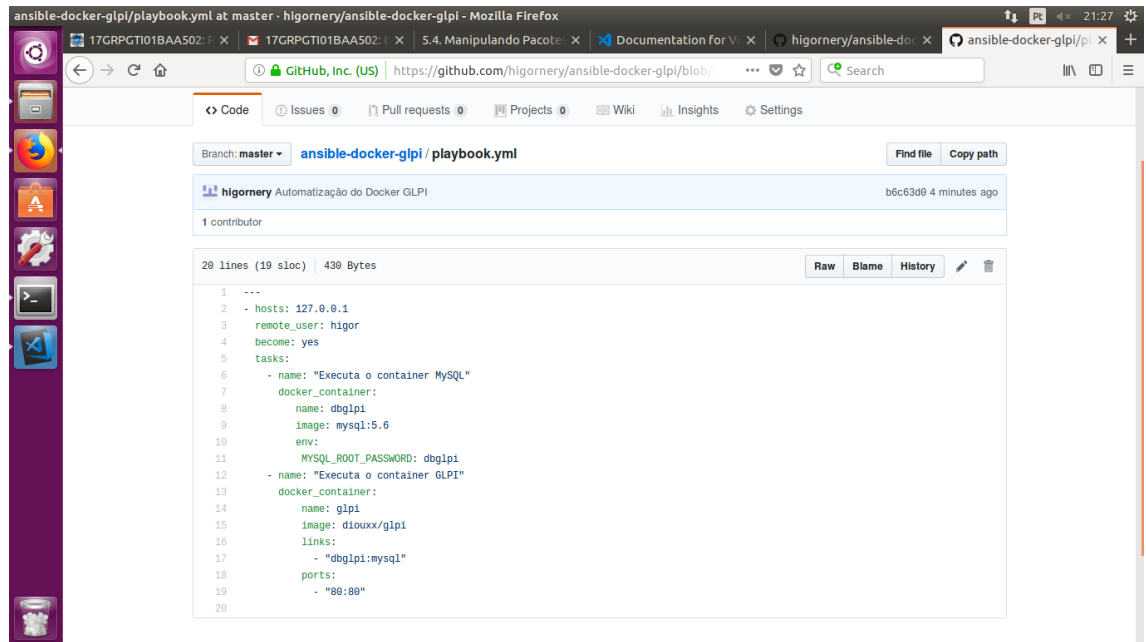
```
playbook.yml - Ansible - Visual Studio Code
EXPLORER
! playbook.yml
ANSIBLE
! playbook.retry
! playbook.yml

3 remote user: higor
4 become: yes
5 tasks:
6   - name: "Executa o container MySQL"
7     docker_container:
8       name: dbglpi
9       image: mysql:5.6
10      env:
11        MYSQL_ROOT_PASSWORD: dbglpi
12  - name: "Executa o container GLPI"
13    docker_container:
14      name: glpi
15      image: dioux/glpi
16      links:
17        - "dbglpi:mysql"
18      ports:
19        - "80:80"
20

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash

PLAY [127.0.0.1] *****
TASK [Gathering Facts] *****
ok: [127.0.0.1]
TASK [Executa o container MySQL] *****
ok: [127.0.0.1]
TASK [Executa o container GLPI] *****
changed: [127.0.0.1]
PLAY RECAP *****
127.0.0.1 : ok=3 changed=1 unreachable=0 failed=0
higor@higor-lab:~/Ansibles
```

- O upload do playbook foi efetuado no repositório no github, no seguinte endereço: <https://github.com/higorner/ansible-docker-glpi.git>.



The screenshot shows a web browser displaying the GitHub repository page for 'ansible-docker-glpi'. The file 'playbook.yml' is selected, showing its content. The file is 29 lines long, 19 sloc, and 430 Bytes. It contains an Ansible playbook with two tasks: 'Executa o container MySQL' and 'Executa o container GLPI'.

```

1 ---
2 - hosts: 127.0.0.1
3   remote_user: higor
4   become: yes
5   tasks:
6     - name: "Executa o container MySQL"
7       docker_container:
8         name: dbglpi
9         image: mysql:5.6
10        env:
11          MYSQL_ROOT_PASSWORD: dbglpi
12    - name: "Executa o container GLPI"
13      docker_container:
14        name: glpi
15        image: dioux/glpi
16        links:
17          - "dbglpi:mysql"
18        ports:
19          - "80:80"
20

```

## 9 PROPOSTA DE SERVIÇOS - TESTE DE PERFORMANCE 6

A partir deste capítulo, será desenvolvida uma proposta comercial baseada nas tecnologias aplicadas nos itens supracitados. Baseado no estudo de caso de uma empresa que precisa tangibilizar sua área de tecnologia da informação como um todo. Entende-se que por meio da aplicação escolhida tal objetivo seja alcançado.



## **9.1 OBJETIVO DA PROPOSTA**

A proposta em questão tem o objetivo de agregar valor de tecnologia a empresa XPTO visando o aumento da produtividade da área de tecnologia da informação.

## **9.2 RESUMO DA PROPOSTA**

Toda a proposta faz parte do planejamento para implementação de uma aplicação para controle de chamados. Foram selecionadas as tecnologias visando a segurança, escalabilidade e alta disponibilidade do ambiente.

## **9.3 DESCRIÇÃO DO CENÁRIO**

A empresa XPTO não dispõe de um sistema de gerenciamento de TI para o seu ambiente. Tal ausência proporciona a intangibilidade da área de tecnológica da informação como um todo.

Gerencialmente a área é um custo para a empresa por não agregar valor tecnológico ao ambiente se tornando um setor de custo alto sem qualquer retorno operacional. Diante do cenário, a diretoria entendeu que uma plataforma de gerenciamento de tecnologia da informação seria o melhor caminho para que o TI pudesse ser transformado em números condizentes com a realidade.

Para que não haja cortes nos custos de pessoal, a empresa não irá realizar altos investimentos a curto prazo para um projeto de implementação de uma solução de fácil acesso que possa gerenciar/ inventariar dispositivos móveis até gerar números de chamados com acompanhamento da qualidade dos mesmos.

## 9.4 SOLUÇÃO PROPOSTA

A solução proposta para o ambiente em questão é a ferramenta open source GLPI, no qual poderemos efetuar a abertura de chamados e juntamente inventariar todo seu parque. Acreditamos que o seu principal objetivo seja a transformação de chamados e atividades de TI em números visando uma decisão estratégica que a direção da empresa terá que tomar perante a área.

Devido ao grau de investimento inicial não ser alto, a melhor opção é realizar a alocação do software em uma nuvem pública. Para a empresa o desembolso será menor e teremos alta qualidade nos serviços seguindo os padrões e normas de segurança com o menor custo

Após um estudo de qualidade seguindo o quadrante mágico do gartner para provedores nuvem publica a escolhida foi a Amazon AWS. No fornecedor teremos maior qualidade, menor custo de implementação e manutenção.

No serviço em questão faremos a implementação, inicialmente, em apenas uma instancia EC2 com containers configurados para que possamos ter menor descapitalização inicial da empresa.

## 9.5 REQUISITOS MÍNIMOS

Para que possamos efetuar a implementação do ambiente, os seguintes requisitos mínimos se fazem necessários:

- Servidor web

**GLPI requer um servidor web que suporta PHP, tais como:**

Apache 2 ou superior (<http://httpd.apache.org>)

Microsoft IIS (<http://www.iis.net>).

PHP

GLPI requer PHP versão 5.3 ou superior (<http://www.php.net>).

- **Extensões PHP necessárias**

As seguintes extensões PHP são necessárias para o bom funcionamento do aplicativo:

JSON: suporte de formato dados estruturados JSON

Mbstring: gerenciamento de cadeia de caracteres de vários bytes

MySQL: conexão com o banco de dados

Sessão: Usuários de suporte de sessão

- **Extensões PHP opcionais, mas recomendadas**

As seguintes extensões são necessárias para recursos opcionais do aplicativo:

CLI: Usando PHP a partir da linha de comando para ações automáticas

CURL: CAS para autenticação

DOMXML: CAS para autenticação

GD: geração de imagens

IMAP: servidores de correio usam para coletar os ingressos ou a autenticação do usuário

LDAP: uso de um diretório externo para autenticação

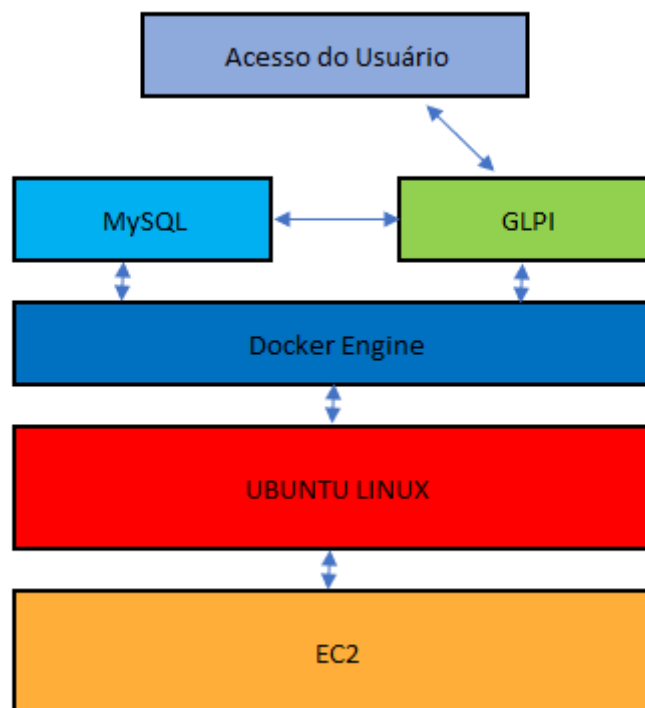
OpenSSL: comunicação criptografada

Tais detalhes técnicos proporcionarão uma ambiente web totalmente disponível e de fácil interação com o usuário.

## 9.6 INFRAESTRUTURA PARA APLICAÇÃO

Conforme supracitado no item 9.4, faremos a implementação, inicialmente, em apenas uma instancia EC2, com containers utilizando a tecnologia Docker e utilizando a automatização da implementação com o recurso chamado Ansible configurados para que possamos ter menor descapitalização inicial da empresa.

Todo controle de versões dos scripts será efetuado via GIT HUB, nos fazendo come que tenhamos maior controle sobre as mudanças.



*Figura 1 - Camadas da arquitetura da aplicação*

## 10 DESCRIÇÃO DA TECNOLOGIA UTILIZADA – TP 7

Visando o melhor custo/ benefício do ambiente faremos a implementação da tecnologia de container Docker, o texto abaixo, de Iago Correa (<http://coral.ufsm.br/pet-si/index.php/o-docker-e-seus-containers-a-nova-era-da-virtualizacao/#more-940>) dá uma definição básica do que é o recurso.

“O Docker é uma plataforma de virtualização Open Source escrita na linguagem Go (linguagem de alto desempenho criada pela Google) e mantida pela Docker Inc (empresa localizada na Califórnia – EUA), que permite a criação e administração de ambientes isolados hospedados em um sistema operacional (SO). A plataforma, permite que seja encapsulado uma aplicação ou, até mesmo um sistema operacional inteiro dentro desses ambientes, que são chamados de containers.”

## 11 COMPARATIVO – TESTE DE PERFORMANCE 7

O docker pode ser implementado em qualquer host de virtualização da atualidade, porém para sua implementação, no projeto da XPTO, viu-se a necessidade de utilizarmos a plataforma EC2, da Amazon.

Para fins de comparação, faremos um paralelo em realizar a implementação on-premisse, virtualizando containers na VMWare, e utilizarmos o recurso na Amazon AWS.

O quadro abaixo mostra a diferença de recursos iniciais para o projeto:

Recursos necessários	VMWARE	EC2
Aquisição de servidores	X	Não se aplica
Aquisição de switches para H.A.	X	Não se aplica
Aquisição de Storage	X	Não se aplica
Custo de implementação	X	Não se aplica
Energia elétrica	X	Não se aplica
Licenciamento	X	Não se aplica
Mensalidade	Não se aplica	X

## 12 PLANO DE IMPLEMENTAÇÃO – TP7 E TP8

Para que haja maior segurança durante o período de implementação, teremos três ambientes para que possamos realizar o trabalho em questão, seguem abaixo seus descritivos.

- Testes – A aplicação será instalada em um servidor, visando a criação de regras aplicadas para o negócio e testaremos novas integrações do sistema
- Homologação - Todos os testes bem-sucedidos serão enviados para a etapa de homologação para que os usuários possam se familiarizar com as novas funcionalidades
- Produção - Por fim, as novas funcionalidades serão aplicadas na produção para que os clientes possam usar no seu dia a dia.

### 12.1 Implementação do EC2 – TP7 e TP8

A implementação se inicia com a criação de uma instancia na Amazon AWS em uma conta de acesso pré configurada para a empresa XPTO. A instancia configurada será a EC2

A sequência de imagens abaixo mostra a configuração da instancia EC2, Ubuntu server na Amazon AWS:

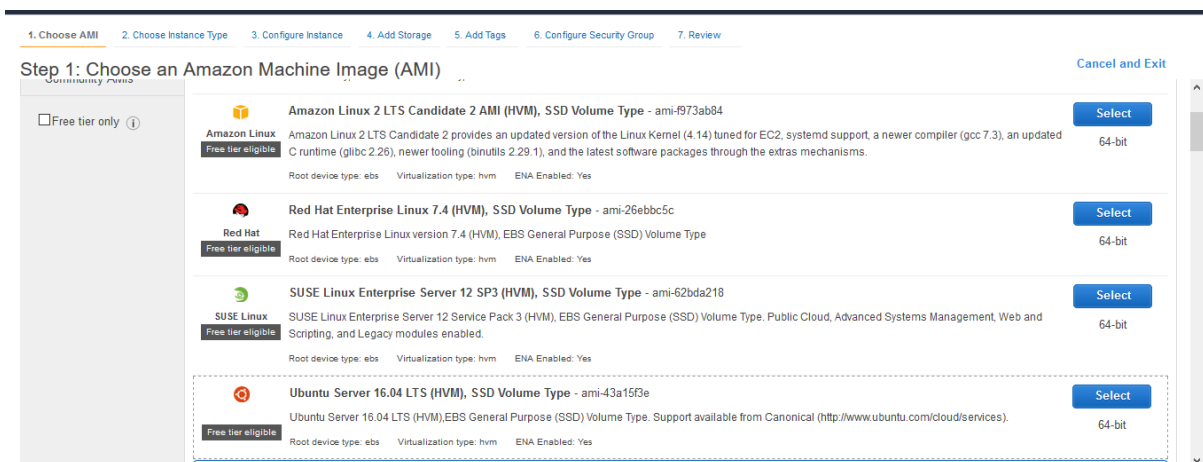


Figura 2 - Seleção da imagem Ubuntu

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

### Step 2: Choose an Instance Type

Currently selected: t2.large (Variable ECUs, 2 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4, 8 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	m5.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
<input type="checkbox"/>	General purpose	m5.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes

Cancel
Previous
Review and Launch
Next: Configure Instance Details

Figura 3 - Seleção do tipo de Imagem

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

### Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0a83a22928e9afbe9	100	General Purpose SSD (GP2)	300 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel
Previous
Review and Launch
Next: Add Tags

Figura 4 - Seleção do tamanho do disco

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

## Step 6: Configure Security Group


A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

**Assign a security group:** ☒ Create a new security group  
☐ Select an existing security group

**Security group name:**

**Description:**

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
SSH ▾	TCP	22	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop	✕
<div>Add Rule</div>					

 **Warning**

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

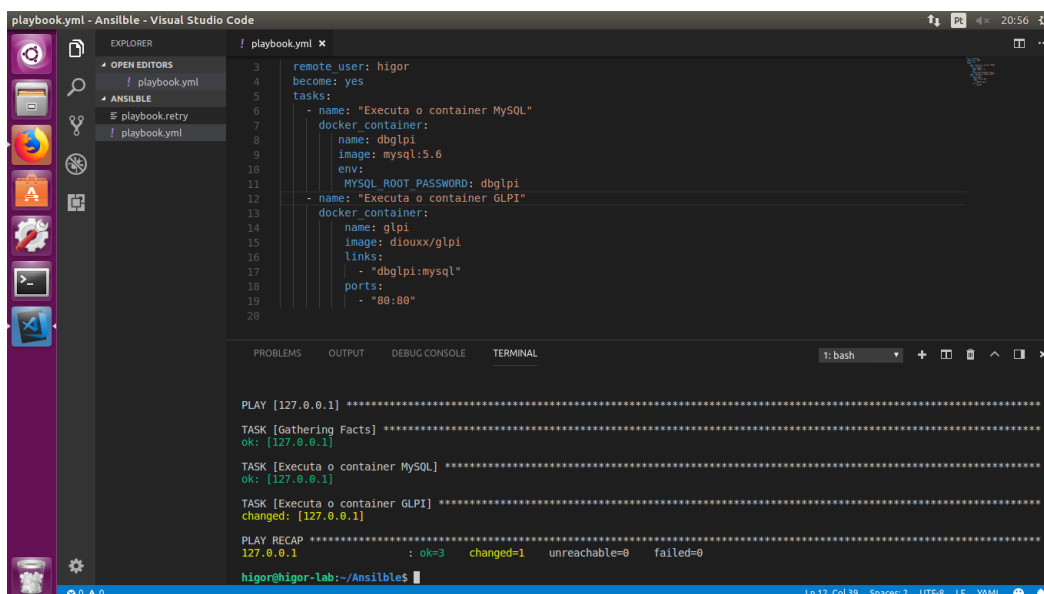
Cancel

Previous

Review and Launch

	XPTO	i-0def5e8341d485f...	t2.large	us-east-1d	pending	Initializing	None	ec2-52-91-28-32.com...	52.91.28.32	-
---	------	----------------------	----------	------------	---	--	------	--	-------------	---

## 12.2 Configuração do Ansible Docker – TP7 e TP8

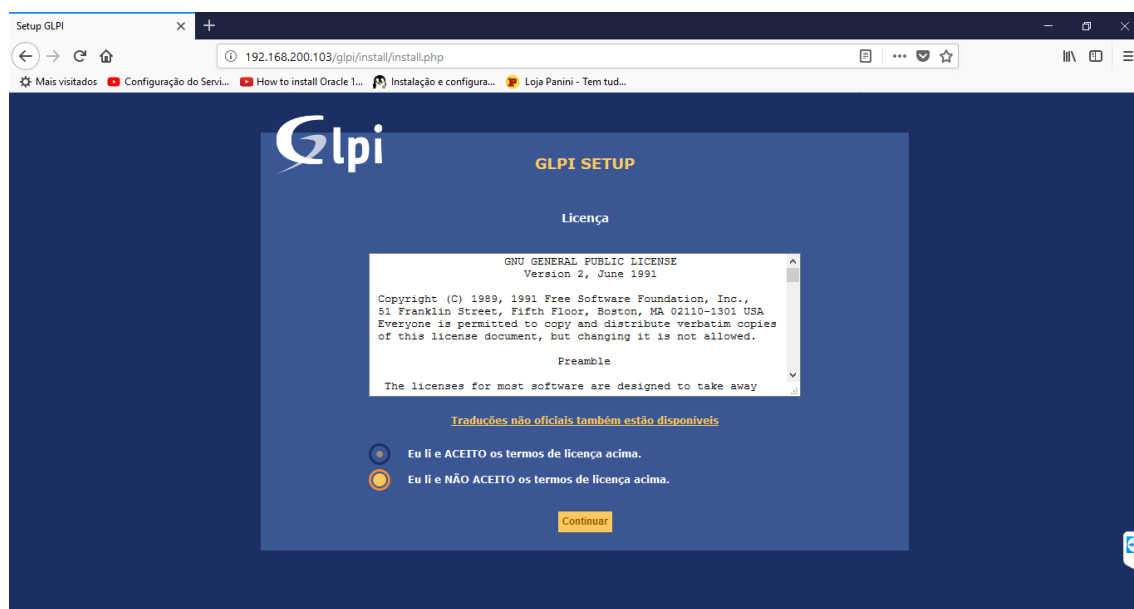




## 13 APLICAÇÃO EM FUNCIONAMENTO – TP8

Após executar as etapas supracitadas, poderemos efetuar a implementação do ambiente em homologação para que os usuários possam simular a abstração de chamados, a área de TI pré cadastrar os ativos até que a aplicação esteja estabilizada.

A imagem abaixo mostra a aplicação em funcionamento:



## 14 CRONOGRAMA DO PROJETO – TP 7

O cronograma do projeto estima cento e dezesseis horas para implementação, a tabela abaixo mostra as atividades relacionadas:

Atividade		Tempo estimado
1	Configurações AWS	
1.1	Criação da instância EC2	01:00:00
1.2	Configurações gerais do Ubuntu	02:00:00
1.3	Instalação das dependencias do docker	01:00:00
1.4.	Instalação das dependencias do Ansible	01:00:00
2	Automatização	
2.1	Criação do playbook	08:00:00
2.2	Execução do playbook docker	03:00:00
3	Regras de negócio	
3.1	Criação das regras de negócio	100:00:00
Total estimado		116:00:00

## 15 CONSIDERAÇÕES SOBRE O PROJETO - TP9

Entendemos que para a implementação do projeto, as 116 horas sejam suficientes para a implementação, pois a montagem das regras de negócio demandará mais tempo.

Os recursos utilizados para implementação do ambiente estão totalmente de acordo com os pré-requisitos solicitados para o que seja obtido sucesso no projeto. As funcionalidades descritas nos itens supracitados, trarão os números que a diretoria necessita para ter números reais sobre a área de tecnologia da informação, bem como a obtenção de maior gerencia sobre o setor afim de redirecionar custos de maneira mais assertiva para TI.

## 15.1 Melhorias sugeridas – TP9

Após a implementação, o ideal é efetuarmos os logins únicos, ou seja, vinculados ao active directory, juntamente com um servidor de redirecionamento e a implementação de um certificado digital wildcard para que a aplicação seja publicada na internet com maior segurança.

## 15.2 Repositório para o projeto

Todas as documentações juntamente com suas alterações do projeto, estarão disponíveis em um repositório do github, no qual também terão o armazenamento dos códigos de configuração do ambiente, juntamente com o seu versionamento.

- Endereço do repositório no github

<https://github.com/higornery/ansible-docker-glpi>

- A imagem abaixo mostra o repositório do git hub no ar, com o playbook inicial do projeto

