



Instituto Infnet

Escola de Negócios

Projeto de Bloco: Arquitetura de Infraestrutura de Aplicações

TP9

Gabriel Guimarães Pires da Silva

Matrícula:170.347.347-71

Rio de Janeiro, 24 de agosto de 2020.

Sumário

Sobre a aplicação	4
Informações sobre como o projeto da aplicação aceita requisições de desenvolvimento e correção de bugs por parte dos usuários e stakeholders.	4
Detalhes técnicos sobre como o código-fonte da aplicação é gerenciado.	4
Informações sobre como compilar/construir os componentes da aplicação e transformá-la em uma unidade instalável.	4
Uma estimativa da infraestrutura e ferramentas de gerenciamento de configuração que podem ser usadas para gerenciar a instalação da aplicação.	6
Um esboço de um plano de implantação da aplicação em sua estrutura corporativa.	7
TP2.....	8
Escreva um playbook Ansible que instale a aplicação escolhida por você para realização do TP1	8
Estrutura Ansible	8
TP3.....	11
1 – Passo: Iniciar um repositório na sua máquina	11
2 – Passo: Adicionando uma Origin.....	11
3 – Enviando alterações para o GitHub.....	12
TP4.....	15
TP5.....	17
TP6 – Início do Projeto	19
O tipo de negócio/processo/problema que pretende tratar a partir de uma aplicação distribuída rodando sobre uma infraestrutura com virtualização.	19
Uma justificativa de porque este problema é relevante.	19
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.	19
Pré-requisitos	20
Uma proposta inicial de como organizar a infraestrutura de sua aplicação, de acordo com o sistema de virtualização em que você pretende implementá-la.	20
TP7.....	21
Uma pequena descrição textual teórica sobre a arquitetura da solução de virtualização que você pretende utilizar.	21
Comparações entre a forma como você pretende implantar seu ambiente e outras abordagens, usando outras ferramentas.....	21
Um planejamento passo a passo (com descrições de cada etapa) de como será feita a implantação da aplicação distribuída virtualizada.....	22
Um cronograma estimado com o prazo para execução de cada atividade.	23

TP8 – Implementação da aplicação.....	24
Referências dos downloads e/ou versões de todos os componentes da solução implementada.	24
Todos os passos/capturas de tela de configuração de sua aplicação.....	24
Uma captura de tela de sua aplicação em funcionamento.	28
TP9.....	29
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).....	29
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.....	29
Se as funcionalidades previstas na solução original funcionaram a contento. Se não foram, explique as razões.	29
Quais seriam as melhorias futuras que poderiam ser feitas no projeto executado, adotando soluções adicionais, aperfeiçoando instalações ou tornando o cenário mais complexo.	30
Bibliografia e Referências.....	31

Sobre a aplicação

A aplicação escolhida é o Zabbix. O Zabbix é uma ferramenta de monitoramento de redes, servidores e serviços, pensada para monitorar a disponibilidade, experiência de usuário e qualidade de serviços.

Todos os relatórios Zabbix e estatísticas, bem como os parâmetros de configuração, são acessados através de uma ferramenta Web que é o front-end do produto. Uma ferramenta web assegura que o status da rede e da saúde dos servidores pode ser avaliado a partir de qualquer localização. Devidamente configurado, Zabbix pode desempenhar um papel importante no controle da infra-estrutura de TI. Isto é igualmente verdadeiro para as pequenas organizações com poucos servidores e para as grandes empresas com muitos servidores.

Informações sobre como o projeto da aplicação aceita requisições de desenvolvimento e correção de bugs por parte dos usuários e stakeholders.

O Zabbix é uma ferramenta gratuita e Open Source isso significa que ele pode ser adaptado a cada modelo de negócio, o que facilita no desenvolvimento de novas funcionalidades e correção de bugs por parte do time de DevOps.

Detalhes técnicos sobre como o código-fonte da aplicação é gerenciado.

O código-fonte do Zabbix é gerenciado através do GitHub. O GitHub é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo. GitHub é amplamente utilizado por programadores para divulgação de seus trabalhos ou para que outros programadores contribuam com o projeto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam repositórios remotos (issues, pull request).

Link do repositório: <https://github.com/zabbix/zabbix>

Informações sobre como compilar/construir os componentes da aplicação e transformá-la em uma unidade instalável.

O Zabbix será executado em um servidor Linux hospedado na nuvem:

1 - Baixar o pacote com o código fonte e escolher uma plataforma:

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	DATABASE	WEB SERVER
4.4	Red Hat Enterprise Linux	10 (Buster)	MySQL	Apache
4.0 LTS	CentOS	9 (Stretch)	PostgreSQL	NGINX
3.0 LTS	Oracle Linux	8 (Jessie)		
Pre 5.0	Ubuntu			
	Debian			
	SUSE Linux Enterprise Server			
	Raspbian			

2 - Instalar e configurar o servidor Zabbix de acordo com a plataforma:

a) Instalar o repositório do Zabbix:

```
# wget
https://repo.zabbix.com/zabbix/4.4/debian/pool/main/z/zabbix-
release/zabbix-release_4.4-1+buster_all.deb
# dpkg -i zabbix-release_4.4-1+buster_all.deb
# apt update
```

b) Instale o servidor Zabbix, front-end, agente

```
# apt install zabbix-server-mysql zabbix-frontend-php zabbix-
apache-conf zabbix-agent
```

c) Criar o banco de dados inicial

```
# mysql -uroot -p
password
mysql> create database zabbix character set utf8 collate
utf8_bin;
mysql> grant all privileges on zabbix.* to zabbix@localhost
identified by 'password';
mysql> quit;
```

No host do servidor Zabbix, importe o esquema e os dados iniciais. Será solicitado a inserir a senha recém-criada.

```
# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz |
mysql -uzabbix -p zabbix
```

d) Configurar o banco de dados para o servidor Zabbix

Editar o arquivo `/etc/zabbix/zabbix_server.conf`

```
DBPassword=password
```

e) Configurar o PHP para front-end do Zabbix

Editar o arquivo `/etc/zabbix/apache.conf`, e definir o fuso horário certo para você.

```
# php_value date.timezone BRT/UTC-3h
```

f) Iniciar os processos do servidor e agente do Zabbix

Inicie os processos do servidor e do agente Zabbix e inicie na inicialização do sistema.

```
# systemctl restart zabbix-server zabbix-agent apache2  
# systemctl enable zabbix-server zabbix-agent apache2
```

g) Configurar o Zabbix front-end

Se conectar ao recém instalado Zabbix: http://server_ip_or_name/zabbix

Seguir as etapas descritas no documento abaixo:

https://www.zabbix.com/documentation/4.4/manual/installation/install#installing_frontend

Uma estimativa da infraestrutura e ferramentas de gerenciamento de configuração que podem ser usadas para gerenciar a instalação da aplicação.

O Zabbix será implantado em 3 servidores dedicados, conforme a imagem abaixo:



- **Zabbix Server:**
OS: Debian 10
Memória RAM: 16Gb
Processador: Intel Core I7
SSD: 500Gb
- **Zabbix Front-End:**
OS: Debian 10
Memória RAM: 16Gb
Processador: Intel Core I7
SSD: 500Gb
- **Zabbix DataBase:**

Oracle MySQL
Capacidade inicial: 5Tb

Um esboço de um plano de implantação da aplicação em sua estrutura corporativa.

O Zabbix será implantado em 3 passos:

- 1 – Adquirir a infraestrutura necessária através dos serviços de computação em nuvem;
- 2 – Fazer a pré-instalação dos sistemas;
- 3 – Instalar e configurar o Zabbix;

TP2

Escreva um playbook Ansible que instale a aplicação escolhida por você para realização do TP1

Estrutura Ansible:

Inventory: é o arquivo que contém a lista de hosts que serão gerenciados pelo Ansible.

Modules: são os recursos (serviços, pacotes, arquivos) que o Ansible usará para controlar e gerenciar os hosts.

Tasks: são as tarefas que serão executadas nos hosts, podemos usar também o modo de task chamado ad-hoc (comandos executados diretamente pela linha de comando sem playbooks).

Playbooks: é um conjunto de tarefas, passos de um processo de configuração que são escritos em YAML (Yet Another Markup Language) que serão executadas nos hosts remotos pelo Ansible. É uma linguagem de fácil interpretação.

Pelo fato da minha playbook se tratar de vários arquivos, o mesmo será disponibilizado no TP3 através do GitHub.

Instalação do Ansible:

Foi utilizado o sistema operacional CentOS.

```
#yum install ansible

gabriel@localhost:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path =
  ['/home/gabriel/.ansible/plugins/modules',
  '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-
  packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.2 (default, Apr 27 2020, 15:53:34) [GCC
  9.3.0]
```



```
gabriel@localhost:/etc/ansible/roles/ZabbixServerAllInOne
File Edit View Search Terminal Help
[root@localhost ZabbixServerAllInOne]# tree
.
├── defaults
│   └── main.yml
├── deploy.yml
├── handlers
│   └── main.yml
├── hosts.ini
├── tasks
│   ├── common.yml
│   ├── main.yml
│   ├── MySQL.yml
│   ├── Zbx-frontend.yml
│   └── Zbx-server.yml
└── templates
    ├── user.root.mycnf.j2
    ├── zabbix_server.conf.44.j2
    └── zabbix_web.conf.j2

4 directories, 12 files
[root@localhost ZabbixServerAllInOne]# ansible-playbook deploy.yml
```

Para instalação da playbook criada é necessário executar o seguinte comando.

```
# ansible-playbook deploy.yml
```

```
TASK [ZabbixServerAllInOne : Create /etc/zabbix/scripts] *****
changed: [10.0.0.240]

TASK [ZabbixServerAllInOne : Make sure zabbix-server service is running] *****
changed: [10.0.0.240]

TASK [ZabbixServerAllInOne : Copy zabbix_server.conf template] *****
changed: [10.0.0.240]

TASK [ZabbixServerAllInOne : Set some fact] *****
ok: [10.0.0.240]

TASK [ZabbixServerAllInOne : Install Zabbix Repo.] *****
ok: [10.0.0.240] => (item={u'state': u'present', u'description': u'Zabbix Official Repository - $basearch', u'pgpcheck': 0, u'pgpkey': u'file:///etc/rpm-gpg/RPM-GPG-KEY-ZABBIX', u'baseurl': u'http://repo.zabbix.com/zabbix/4.4/rhel/7/$basearch/', u'name': u'zabbix'})
ok: [10.0.0.240] => (item={u'state': u'present', u'description': u'Zabbix Official Repository non-supported - $basearch', u'pgpcheck': 0, u'pgpkey': u'file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX', u'baseurl': u'http://repo.zabbix.com/non-supported/rhel/7/$basearch/', u'name': u'zabbix-supported'})

TASK [ZabbixServerAllInOne : Install package Zabbix] *****
changed: [10.0.0.240]

TASK [ZabbixServerAllInOne : Make sure httpd service is running] *****
changed: [10.0.0.240]

TASK [ZabbixServerAllInOne : Copy web.zabbix.conf template] *****
changed: [10.0.0.240]

TASK [ZabbixServerAllInOne : Create firewall rule] *****
changed: [10.0.0.240]

TASK [ZabbixServerAllInOne : Firewalld reload] *****
changed: [10.0.0.240]

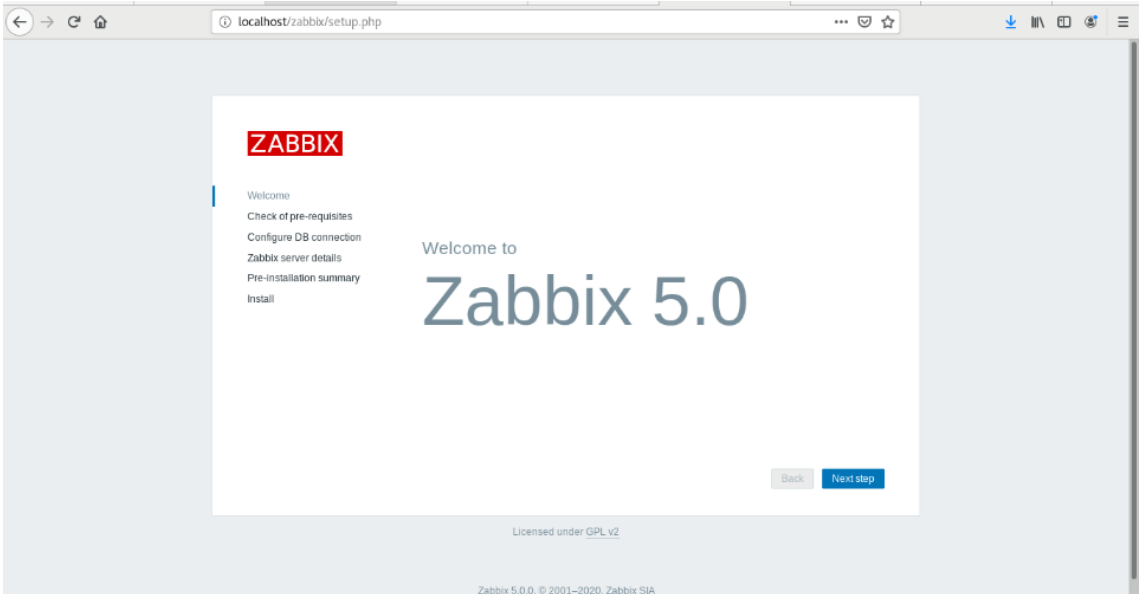
RUNNING HANDLER [ZabbixServerAllInOne : restart mysql] *****
changed: [10.0.0.240]

RUNNING HANDLER [ZabbixServerAllInOne : restart zabbix-server] *****
changed: [10.0.0.240]

RUNNING HANDLER [ZabbixServerAllInOne : restart httpd] *****
changed: [10.0.0.240]

PLAY RECAP *****
10.0.0.240 : ok=38 changed=26 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0
```

Após a conclusão da instalação já é possível acessar a interface Web do Zabbix.

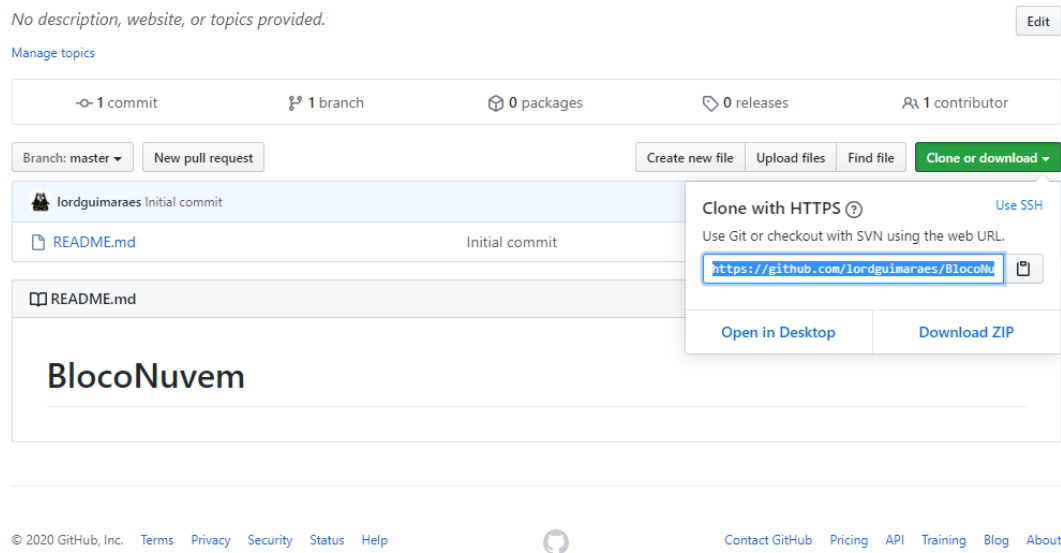


TP3

A ferramenta utilizada para manipulação do repositório será o GitHub, pois tenho preferências por linha de comando.

1 – Passo: Iniciar um repositório na sua máquina

Após a criação do repositório no site é necessário seguir os seguintes passos.



1.1 - Abri o GitBash e digitar o seguinte comando para fazer o download do repositório:

```
$ git clone https://github.com/lordguimaraes/BlocoNuvem.git
Cloning into 'BlocoNuvem'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
unpacking objects: 100% (3/3), 610 bytes | 14.00 KiB/s, done.
```

1.2 Transformar a pasta em um repositório Git, com o seguinte comando:

```
$ git init
Reinitialized existing Git repository in
C:/Users/gaguimaraes/Documents/Facul/Nuvem/BlocoNuvem/.git/
```

2 – Passo: Adicionando uma Origin

2.1 - Vamos agora executar esse comando:

```
git remote add origin https://github.com/lordguimaraes/BlocoNuvem.git
```

Basicamente o que estamos dizendo nessa linha de comando é:

"Git, esse meu repositório local se conectará com um remoto, o caminho dele(origin) é git@github.com:lordguimaraes/ BlocoNuvem.git. Estabeleça essa comunicação pra mim".

3 – Enviando alterações para o GitHub

Para saber as alterações realizadas é necessário digitar o seguinte comando:

```
$ git status
```

Ele irá mostrar

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        defaults/
        deploy.yml
        handlers/
        hosts.ini
        tasks/
        templates/

nothing added to commit but untracked files present (use "git add" to track)
```

Ele está dizendo aí que os arquivos dessa pasta está untracked, ou seja, não está sendo monitorado/gerenciado pelo Git. Então iremos adicionar os arquivos ao GitHub com o seguinte comando:

```
$ git add .
```

Output

```
$ git add .
warning: LF will be replaced by CRLF in defaults/main.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in deploy.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in handlers/main.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in hosts.ini.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tasks/MySQL.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tasks/Zbx-frontend.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tasks/Zbx-server.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tasks/common.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tasks/main.yml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in templates/user.root.mycnf.j2.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in templates/zabbix_server.conf.44.j2.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in templates/zabbix_web.conf.j2.
The file will have its original line endings in your working directory
```

Com esse comando os arquivos foram adicionados ao Git, isso é visto através do status:

```
$ git status
```

```
On branch master
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   defaults/main.yml
    new file:   deploy.yml
    new file:   handlers/main.yml
    new file:   hosts.ini
    new file:   tasks/MySQL.yml
    new file:   tasks/Zbx-frontend.yml
    new file:   tasks/Zbx-server.yml
    new file:   tasks/common.yml
    new file:   tasks/main.yml
    new file:   templates/user.root.mycnf.j2
    new file:   templates/zabbix_server.conf.44.j2
    new file:   templates/zabbix_web.conf.j2
```

Agora ele me mostra que o arquivo está para ser comitado, para isso utilizaremos o seguinte comando.

```
$ git commit -m "primeiro commit tp3"
```

Output:

```
[master (root-commit) 14ae59d] tp3 - bloco arquitetura e nuvem
Committer: Gabriel Guimarães <gabriel.dsilva@al.infnet.edu.br>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
12 files changed, 895 insertions(+)
create mode 100644 defaults/main.yml
create mode 100644 deploy.yml
create mode 100644 handlers/main.yml
create mode 100644 hosts.ini
create mode 100644 tasks/MySQL.yml
create mode 100644 tasks/Zbx-frontend.yml
create mode 100644 tasks/Zbx-server.yml
create mode 100644 tasks/common.yml
create mode 100644 tasks/main.yml
create mode 100644 templates/user.root.mycnf.j2
create mode 100644 templates/zabbix_server.conf.44.j2
create mode 100644 templates/zabbix_web.conf.j2
```

Por últimos iremos enviar os arquivos para o repositório público criado.

```
$ git push origin master
```

Output:

```
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 8.57 KiB | 1.07 MiB/s, done.
```

```
Total 18 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/lordguimaraes/BlocoNuvem
 9da843a..48de062  master -> master
```

Com isso os arquivos foram enviados para o nosso repositório e é possível visualizar através do link: <https://github.com/lordguimaraes/BlocoNuvem>

2 commits		1 branch	0 packages	0 releases	1 contributor
Branch: master	New pull request	Create new file	Upload files	Find file	Clone or download
lordguimaraes primeiro commit tp3		Latest commit 48de062 5 minutes ago			
defaults	primeiro commit tp3	5 minutes ago			
handlers	primeiro commit tp3	5 minutes ago			
tasks	primeiro commit tp3	5 minutes ago			
templates	primeiro commit tp3	5 minutes ago			
README.md	Initial commit	1 hour ago			
deploy.yml	primeiro commit tp3	5 minutes ago			
hosts.ini	primeiro commit tp3	5 minutes ago			

TP4

Nessa etapa do projeto de bloco a aplicação para ser instalada na Docker foi o MYSQL, que no futuro será utilizada em conjunto com o Zabbix.

Existem duas maneiras de instalar a Docker, a primeira é a Docker integrada a VMware que exige uma infraestrutura mais robusta, por outro lado temos a segunda opção que é a Docker baseada em SO onde é possível isolar as aplicações com os recursos já existentes, portanto, a Docker baseada em SO foi a solução escolhida.

O utilitário dnf config-manager nos permite, entre outras coisas, habilitar ou desabilitar facilmente um repositório em nossa distribuição. Por padrão, apenas os repositórios appstream e baseos são ativados no RHEL8; precisamos adicionar e ativar também o repositório docker-ce. Tudo o que precisamos fazer para realizar esta tarefa é executar o seguinte comando:

```
# dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
```

Agora, usaremos o comando abaixo do dnf para instalar a versão mais recente do Docker

```
# dnf install docker-ce --nobest -y
```

Após a instalação do docker, iremos iniciar e habilitar serviço usando os seguintes comandos

```
# systemctl start Docker
# systemctl enable docker
```

Com a docker instalada precisamos criar uma instância com a versão mais recente estável do MySQL com o seguinte comando

```
# docker run --name container-teste-mysql -e
MYSQL_ROOT_PASSWORD=senha-mysql -d mysql/mysql-server:latest
```

É possível ver os containers que estão em execução com o seguinte comando

```
# docker ps
```

```
root@localhost:/home/gabriel
File Edit View Search Terminal Help
[root@localhost gabriel]# docker ps
CONTAINER ID        STATUS          IMAGE                        PORTS                COMMAND              NAMES                CREATE
4d2eebca3350        Up 37 seconds (healthy)    mysql/mysql-server:latest    3306/tcp, 33060/tcp  "/entrypoint.sh mysq...  container-teste-mysq  38 seconds ago
```

Agora podemos acessar a Shell do container e testar o MySQL instalado com o seguinte comando

```
# docker exec -it container-teste-mysql bash
# mysql -uroot -p
```

```
root@localhost:/home/gabriel
File Edit View Search Terminal Help
[root@localhost gabriel]# docker exec -it container-teste-mysql bash
bash-4.2# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.20 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

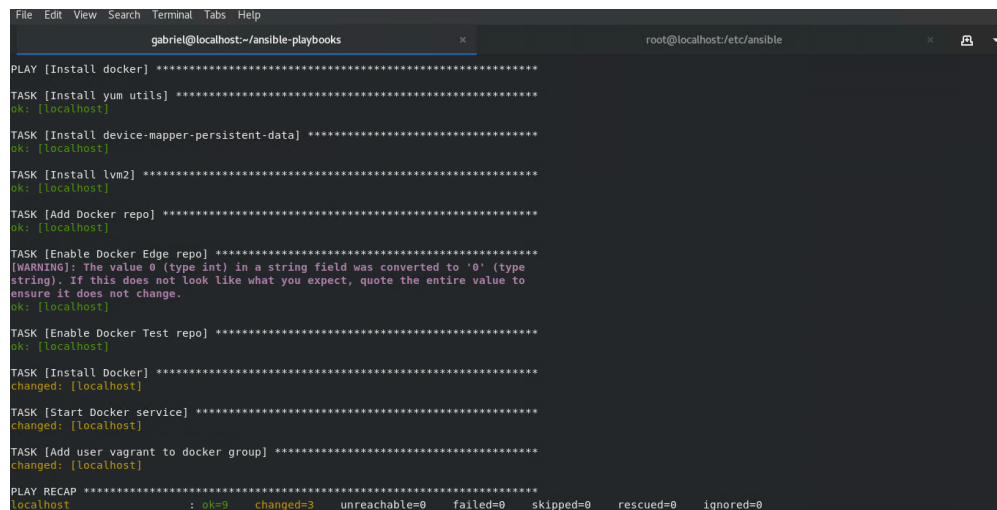
Com isso concluímos a instalação de uma solução de container e a instalação de uma aplicação dentro desse container.

TP5

Nesse trabalho não está muito específico se a playbook a ser criada precisa ser referente ao projeto final, portanto, para esse TP a playbook criada será para a instalação do Docker. O Docker tem como objetivo criar, testar e implementar aplicações em um ambiente separado da máquina original, chamado de container. Dessa forma, o desenvolvedor consegue empacotar o software de maneira padronizada. Isso ocorre porque a plataforma disponibiliza funções básicas para sua execução, como: código, bibliotecas, runtime e ferramentas do sistema.

Playbook: <https://gist.github.com/lordguimaraes/85fed8812d03a8e0819cdc39310857e0>

Instalação da playbook:



```
File Edit View Search Terminal Tabs Help
gabriel@localhost:~/ansible-playbooks x root@localhost:/etc/ansible x
PLAY [Install docker] *****
TASK [Install yum utils] *****
ok: [localhost]
TASK [Install device-mapper-persistent-data] *****
ok: [localhost]
TASK [Install lvm2] *****
ok: [localhost]
TASK [Add Docker repo] *****
ok: [localhost]
TASK [Enable Docker Edge repo] *****
(WARNING): The value 0 (type int) in a string field was converted to '0' (type
string). If this does not look like what you expect, quote the entire value to
ensure it does not change.
ok: [localhost]
TASK [Enable Docker Test repo] *****
ok: [localhost]
TASK [Install Docker] *****
changed: [localhost]
TASK [Start Docker service] *****
changed: [localhost]
TASK [Add user vagrant to docker group] *****
changed: [localhost]
PLAY RECAP *****
localhost : ok=9  changed=3  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

Resultado pós instalação:

```
root@localhost:/home/gabriel/ansible-playbooks
File Edit View Search Terminal Tabs Help
root@localhost:/home/gabriel/ansibl... x root@localhost:/etc/ansible x
[root@localhost ansible-playbooks]# docker -v
Docker version 19.03.12, build 48a66213fe
[root@localhost ansible-playbooks]#
```

TP6 – Início do Projeto

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

As empresas no geral são muito dependentes de seus sistemas que visam aperfeiçoar os trabalhos e serviços dentro de uma organização. Nesse ponto percebe-se a necessidade e importância de manter os serviços funcionando corretamente. Na TI sabemos que não é muito difícil de sistemas e hardwares apresentarem falhas, portanto, para tentar prevenir esse tipo de acontecimento e garantir a continuidade do negócio precisamos criar o monitoramento de rede, que deve acontecer de forma constante e pro ativa.

Uma justificativa de porque este problema é relevante.

O ato de monitorar é o mesmo que observar o desempenho de um sistema. Uma forma, inclusive, de manter prevenido para possíveis problemas ou perceber sinais importantes que aponte para um estado de não normalidade do sistema. Com isso será possível tomar medidas de remediação. Isso é chamado de monitoramento proativo e reativo. Um que visa antecipar os problemas e o outro que analisa a melhor forma de resolver os problemas após detectá-los.

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.

A aplicação escolhida é o Zabbix. O Zabbix é uma ferramenta de monitoramento de redes, servidores e serviços, pensada para monitorar a disponibilidade, experiência de usuário e qualidade de serviços.

Todos os relatórios Zabbix e estatísticas, bem como os parâmetros de configuração, são acessados através de uma ferramenta Web que é o front-end do produto. Uma ferramenta web assegura que o status da rede e da saúde dos servidores pode ser avaliado a partir de qualquer localização. Devidamente configurado, Zabbix pode desempenhar um papel importante no controle da infra-estrutura de TI. Isto é igualmente verdadeiro para as pequenas organizações com poucos servidores e para as grandes empresas com muitos servidores.

O Zabbix é uma ferramenta gratuita e Open Source isso significa que ele pode ser adaptado a cada modelo de negócio, o que facilita no desenvolvimento de novas funcionalidades e correção de bugs por parte do time de DevOps.

Pré-requisitos

Memória

Para instalar o Zabbix existem requisitos de memória (128MB) e de armazenamento (256MB disponíveis em disco). Entretanto, a quantidade de memória e de disco, obviamente, dependerá da quantidade de hosts e de parâmetros monitorados.

CPU

O Zabbix Server e, especialmente, seu banco de dados pode exigir quantidade significativa de recursos da CPU dependendo da quantidade de parâmetros monitorados e da engine do SGDB.

Recursos a serem utilizados para esse projeto:

<i>Sistema Operacional</i>	CentOS
<i>Memória RAM</i>	6Gb
<i>Processador</i>	4 núcleos
<i>HD</i>	100Gb
<i>SGDB</i>	MySQL

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

O projeto será executado dentro de uma plataforma de virtualização, mais conhecido como Hypervisor. A principal função da Hypervisor está na execução de ambientes isolados de cada máquina virtual e no gerenciamento do acesso entre os sistemas operacionais convidados (guests) que estão executando em máquinas virtuais e os recursos de hardware do computador (host).

TP7

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

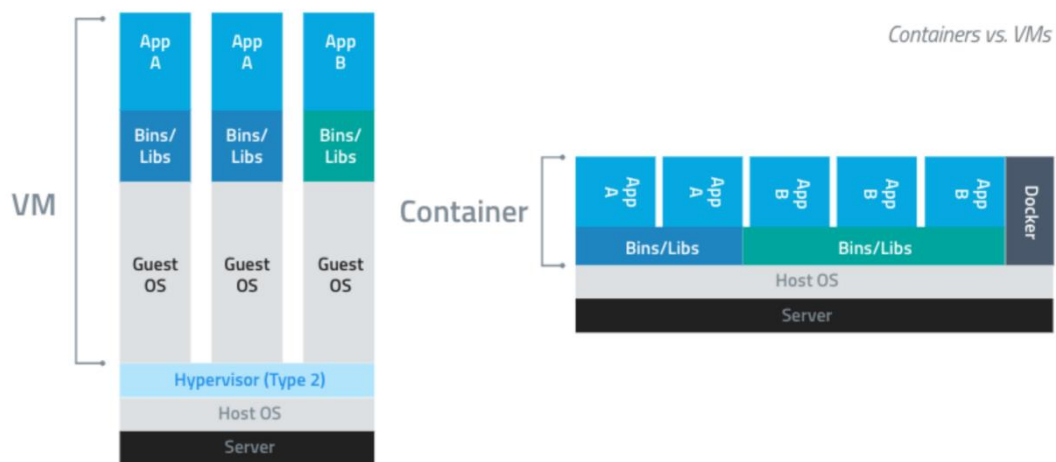
A solução de Hypervisor escolhida foi a VMware. A VMware proporciona integração entre diferentes sistemas operacionais, simulando redes com várias versões do Linux e do Windows. Ele é uma opção para se testar (sem a necessidade de um computador dedicado a esta tarefa) outros sistemas operacionais e aplicativos de forma mais simples, ou até mesmo para manter a compatibilidade de sistemas operacionais e aplicativos antigos (legados) e recentes. Algumas de suas características são:

- Suporta sistemas operando na tecnologia 64 bits, incluindo Windows e Linux;
- Monitora e fiscaliza a infraestrutura em um console central de gerência;
- Tem suporte a dois processadores SMP virtual (SMP virtual é um módulo adicional ao VMware que possibilita que uma única máquina virtual possua mais de 1 processador).

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

Docker vs VMware: A VMware emula o hardware da máquina, enquanto o Docker emula o sistema operacional no qual seu aplicativo é executado. Docker é uma tecnologia de virtualização muito mais leve, já que não precisa emular recursos de hardware de servidor. O foco está em abstrair o ambiente exigido pelo aplicativo, em vez do servidor físico. O VMware, assim como o hardware real da máquina, permite instalar sistemas operacionais e outras tarefas que exigem um servidor completo.

Simplificando, os contêineres fornecem isolamento de processo no nível do sistema operacional, enquanto as máquinas virtuais oferecem isolamento na camada de abstração de hardware (ou seja, virtualização de hardware). Portanto, em casos de uso de IaaS, a virtualização de máquina é ideal, enquanto os contêineres são mais adequados para empacotar / enviar software portátil e modular.



As duas tecnologias podem ser usadas em conjunto para obter benefícios adicionais - por exemplo, contêineres Docker podem ser criados dentro de VMs para tornar uma solução ultra-portátil.

Zabbix + Cloud: Também é possível implementar a solução de Zabbix a partir de uma infraestrutura virtual, inclusive o Zabbix já dispõe de imagens prontas para isso e que possui uma fácil instalação.

A solução de Nuvem é interessante em casos que o projeto terá uma escalabilidade maior, pois realizar upgrades em ambientes de Nuvem é fácil e rápido e você pode aumentar o poder computacional de acordo com as necessidades, diferente de uma infraestrutura própria no qual é necessário planejar bem a demanda para não faltar e também não sobrar recursos.

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

1 Etapa – Adquirir uma solução de virtualização e instalá-lo: O primeiro passo para montar a infraestrutura planejada para esse projeto é adquirir uma solução de virtualização, para o nosso projeto a solução adquirida foi a Vmware.

2 Etapa – Instalar o Sistema Operacional: Após a instalação da Vmware será necessário instalar o Sistema Operacional no qual o projeto será executado, no nosso caso o sistema operacional a ser utilizado será o CentOS.

3 Etapa – Instalação da aplicação: Com o sistema operacional instalado já é possível começar a instalação e configuração da aplicação escolhida.

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

Ação	Status	Início	Término
Adquirir uma solução de virtualização e instalá-lo	Concluída	1/4/2020	27/04/2020
Instalar o Sistema Operacional	Concluída	25/05/2020	25/05/2020
Instalar e configurar a aplicação	Em progresso	8/6/2020	31/08/2020

TP8 – Implementação da aplicação

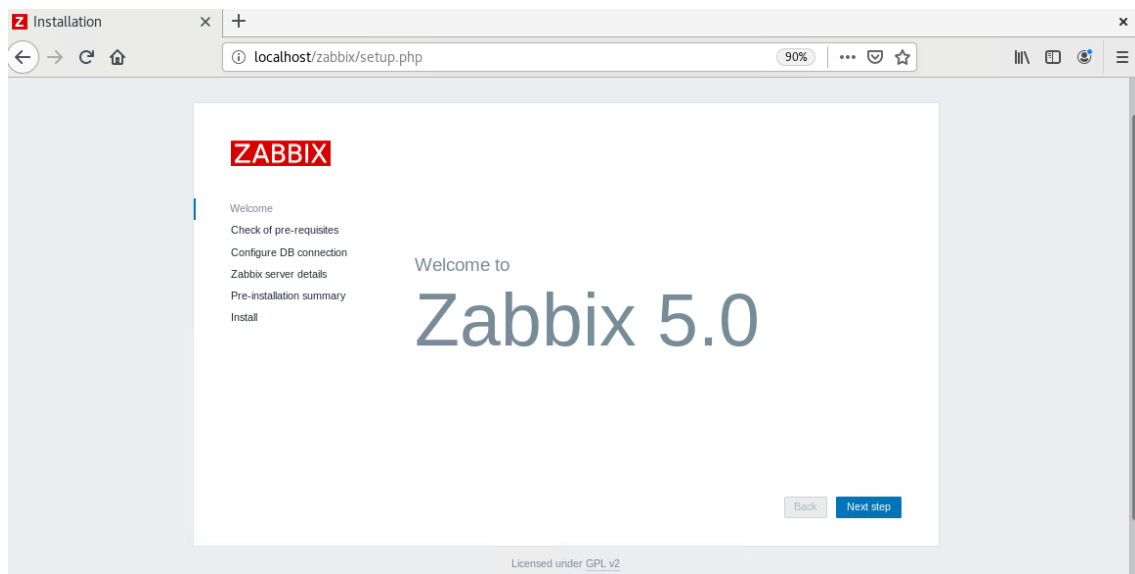
Referências dos downloads e/ou versões de todos os componentes da solução implementada.

1. https://www.zabbix.com/download?zabbix=5.0&os_distribution=centos&os_version=8&db=mysql&ws=apache
2. <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>

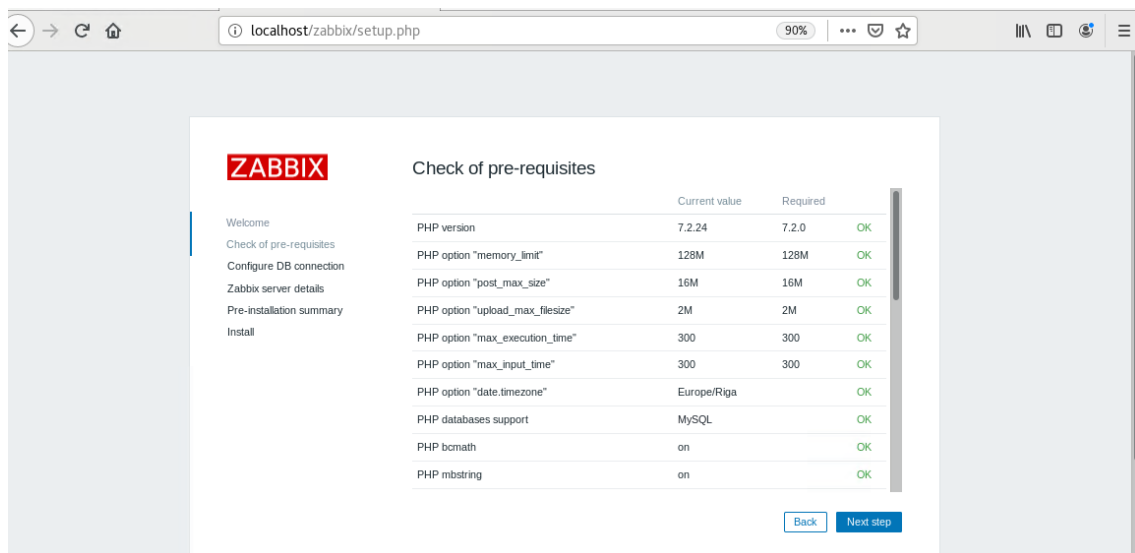
Todos os passos/capturas de tela de configuração de sua aplicação.

Após a instalação do Zabbix precisamos acessar via browser para iniciar as configurações.

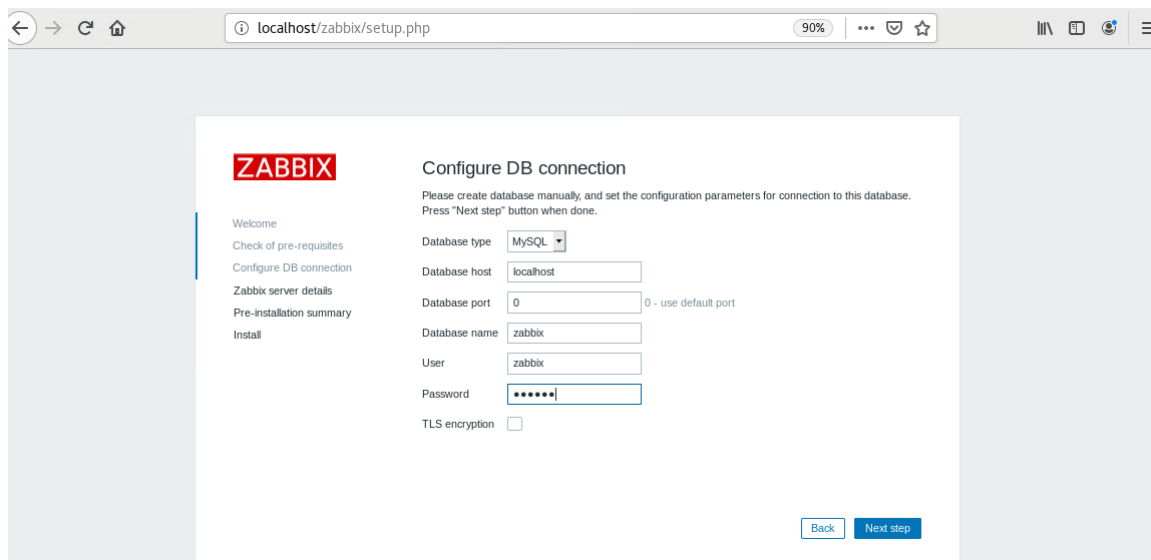
Endereço: <http://localhost/zabbix/>



Esta etapa verifica se todos os pré-requisitos foram atendidos para a instalação do Zabbix. Se houver algum problema, ele precisa ser corrigido. Como não tivemos nenhum erro, vamos para a próxima etapa.



Nessa etapa precisamos inserir o nome e senha do banco de dados criado.



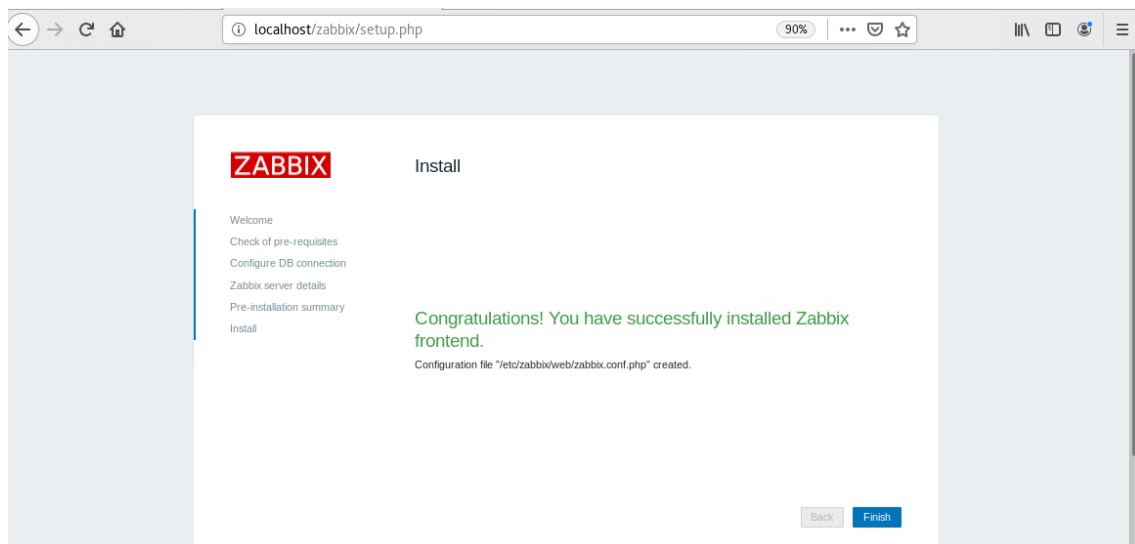
Nessa etapa precisamos informar os detalhes do servidor Zabbix, deixaremos as informações como default.

The screenshot shows a web browser window with the address bar displaying 'localhost/zabbix/setup.php'. The page features the ZABBIX logo and a sidebar with navigation links: Welcome, Check of pre-requisites, Configure DB connection, Zabbix server details (highlighted), Pre-installation summary, and Install. The main content area is titled 'Zabbix server details' and includes a sub-header 'Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional)'. Below this, there are three input fields: 'Host' with 'localhost', 'Port' with '10051', and 'Name' which is empty. At the bottom right, there are 'Back' and 'Next step' buttons.

Aqui será mostrado um resumo das configurações realizadas.

The screenshot shows the 'Pre-installation summary' step of the Zabbix setup. The sidebar navigation is the same as the previous step, with 'Pre-installation summary' highlighted. The main content area is titled 'Pre-installation summary' and includes a sub-header 'Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.' Below this, the configuration parameters are listed in two columns. The first column lists the parameters: Database type, Database server, Database port, Database name, Database user, Database password, TLS encryption, Zabbix server, Zabbix server port, and Zabbix server name. The second column shows the values: MySQL, localhost, default, zabbix, zabbix, *****, false, localhost, 10051, and an empty field. At the bottom right, there are 'Back' and 'Next step' buttons.

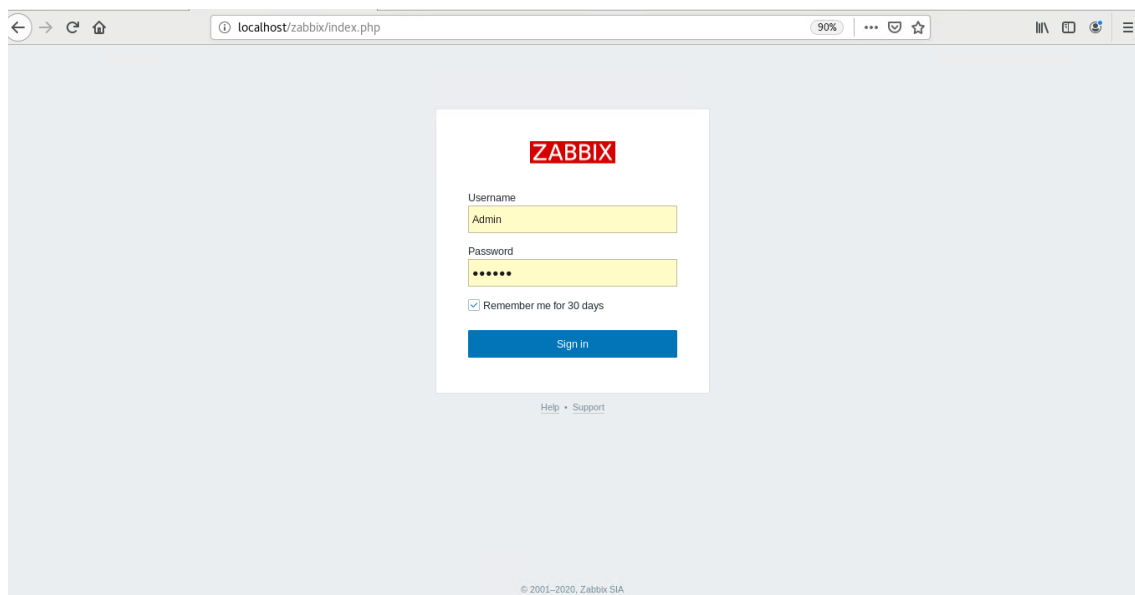
Por último temos a configuração do Zabbix concluída.



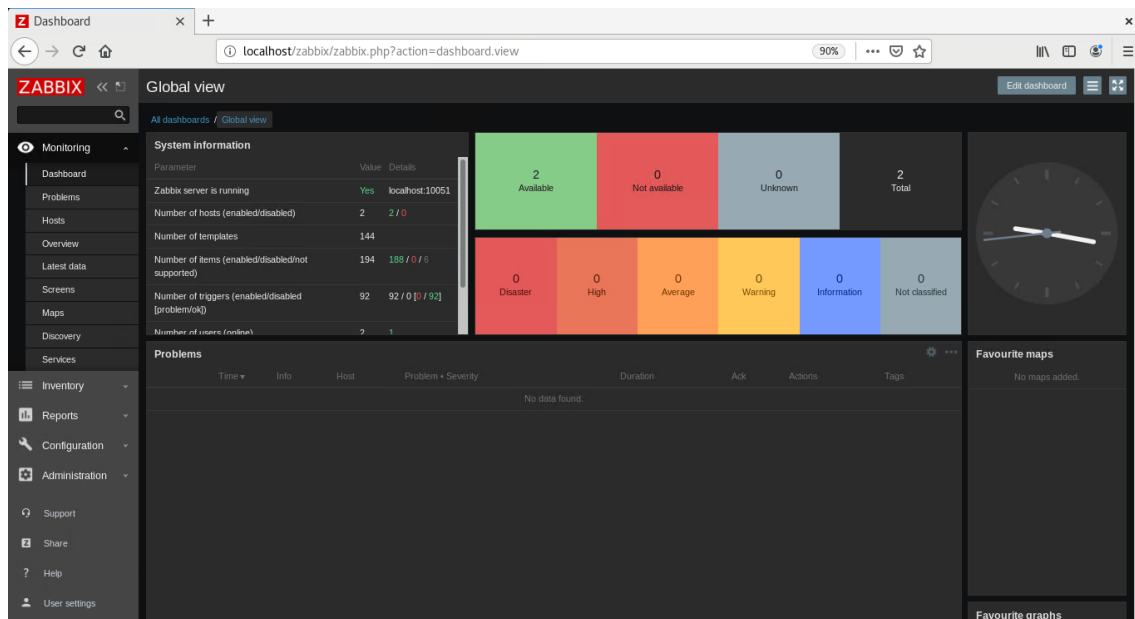
Por fim estaremos na página de login, por default do Zabbix precisamos logar com os seguintes dados:

Usuário: Admin

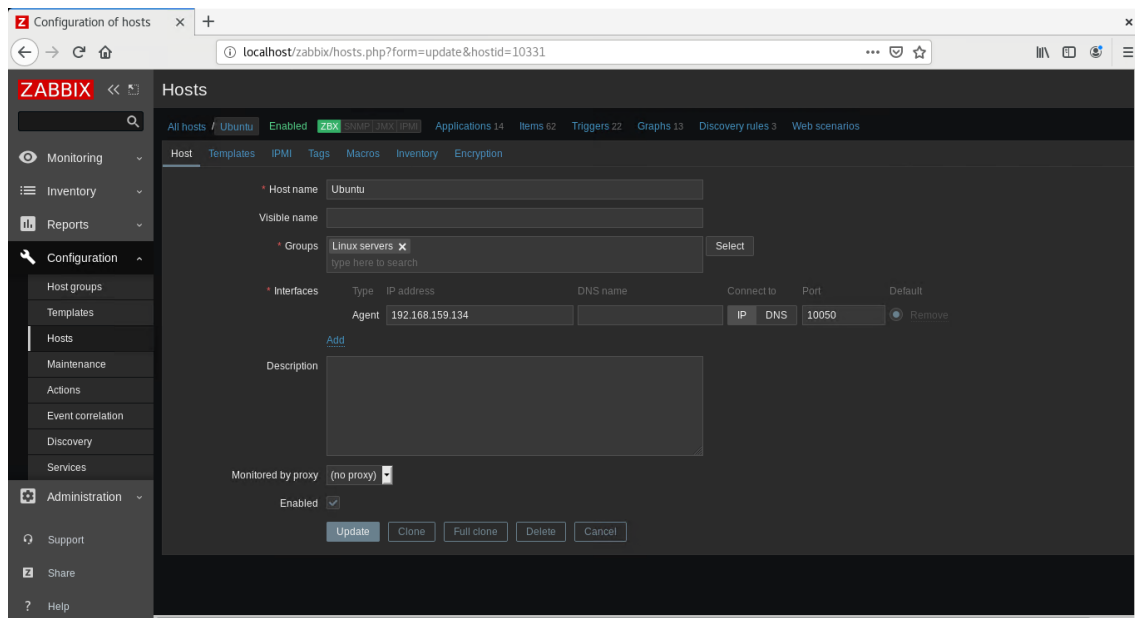
Password: zabbix



Uma captura de tela de sua aplicação em funcionamento.



Após a instalação do Zabbix eu adicionei um Host no qual eu desejo monitorar, nesse caso monitorarei uma máquina virtual Ubuntu.



TP9

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).

Sim, o prazo estabelecido foi o suficiente pois se trata de uma aplicação simples e de fácil implementação.

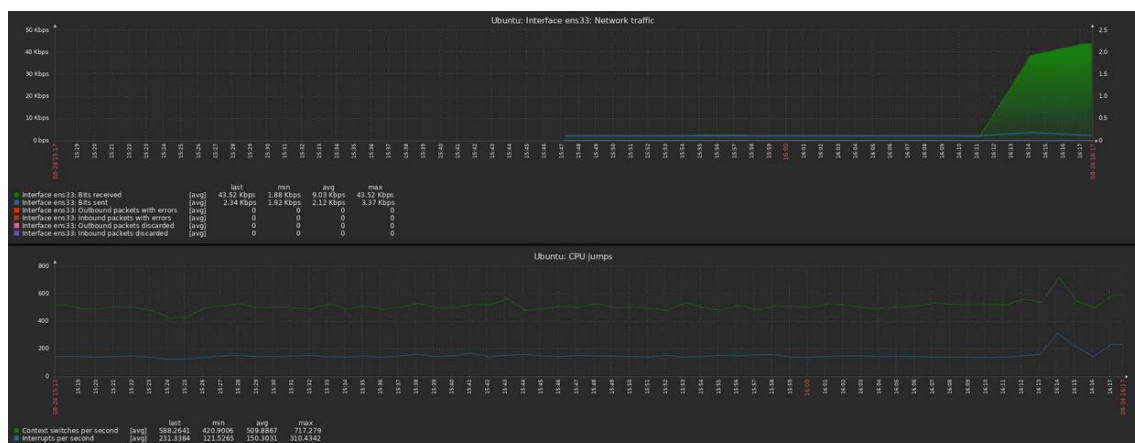
Se os recursos planejados (quantidade de memória, disco, etc.) foram suficientes para colocar a solução em funcionamento. Se não foram, explique as razões e sugira uma nova configuração.

Os recursos planejados e implementados foram suficientes para a execução correta do projeto.

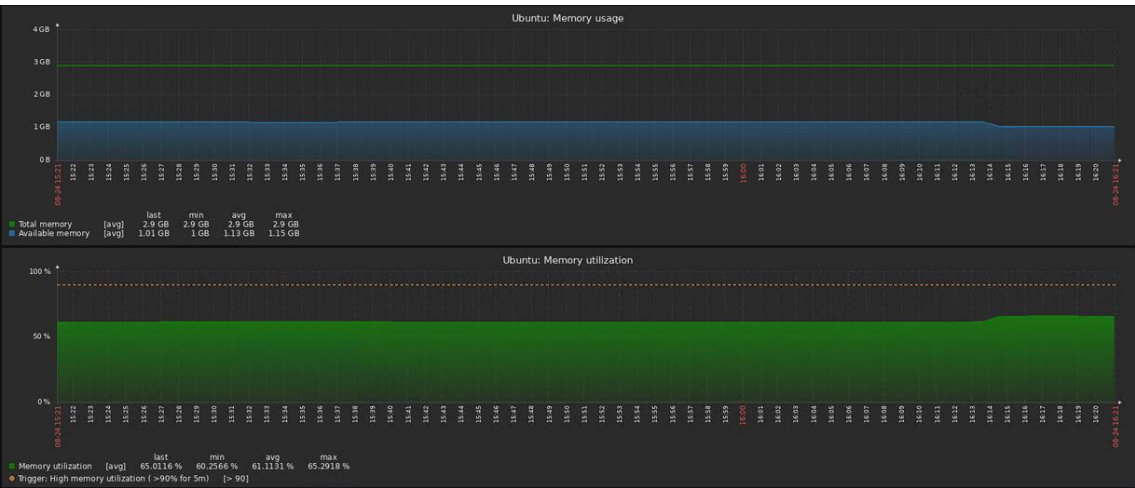
Se as funcionalidades previstas na solução original funcionaram a contento. Se não foram, explique as razões.

A proposta do Zabbix é monitorar ativos em uma rede. Como exemplo e para validar o correto funcionamento da aplicação eu criei uma máquina virtual para testes, e como visto a aplicação teve sucesso em realizar seu trabalho.

Exemplo do gráfico de monitoração de consumo de rede e CPU:



Exemplo de gráfico de utilização de memória:



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

Esse projeto foi todo implementado utilizando uma infraestrutura de virtualização de máquinas virtuais, pois eu sabia exatamente os recursos e quais configurações utilizar para o correto funcionamento da aplicação.

Em casos de projeto com grande escalabilidade e que no futuro o número de ativos a serem monitorados podem crescer demasiadamente, a melhor solução é utilizar a aplicação em nuvem. A aplicação em nuvem permite realizar upgrades de forma simples e rápida e de acordo com a sua demanda. Portanto, uma melhoria futura para essa aplicação seria migrá-la para a Nuvem.

Uma possível melhoria também seria migrar a aplicação para uma solução de container, onde teríamos uma isolação da aplicação em nível de sistema operacional, no entanto, poderia ser um pouco mais difícil de realizar upgrades caso a estrutura venha a crescer.

Bibliografia e Referências

UNIREDE- Disponível em < <https://www.unirede.net/zabbix-4-4-instalacao-modular-da-solucao-de-monitoramento/>>. **ACESSO EM 25/04/2020.**

ZABBIX - Disponível em < https://www.zabbix.com/documentation/1.8/pt/manual/sobre/introducao_ao_zabbix>. **ACESSO EM 25/04/2020.**

MEDIUM- Disponível em < <https://medium.com/@gkal19/enviar-arquivos-para-o-github-fd1f3c78e2eb>>. **ACESSO EM 08/06/2020.**

ESCOLA LINUX- Disponível em <<https://www.escolalinux.com.br/blog/a-importancia-do-monitoramento-proativo-de-redes>>. **ACESSO EM 10/08/2020.**

DEVMEDIA- Disponível em <<https://www.devmedia.com.br/virtualizacao-com-o-hyper-v-artigo-revista-infra-magazine-1/20821>>. **ACESSO EM 10/08/2020.**

UPGUARD- Disponível em <<https://www.upguard.com/blog/docker-vs-vmware-how-do-they-stack-up#:~:text=Docker%20containers%20are%20generally%20faster,%E2%80%94namely%2C%20security%20and%20isolation.&text=So%20for%20application%2Fsoftware%20portability,greate%20isolation%2C%20go%20with%20VMware>>. **ACESSO EM 15/08/2020.**