

MTSDS

Instalação e configuração dos serviços



Conteúdo

Introdução	3
Instalação da Máquina Vagrant com cluster Kubernetes	4
Instalação dos serviços no minikube.....	9
Instalação do front-end.....	11

Introdução

Neste documento serão apresentadas duas formas de instalar os serviços desenvolvidos pelo grupo. A primeira forma é através de uma máquina virtual Vagrant em que foi instalado o kubernetes. A segunda será através do Minikube.

Vantagem de usar a máquina Vagrant:

- É independente do sistema operativo que esteja a ser utilizado.

Desvantagem de usar a máquina Vagrant:

- Exige bastantes recursos de *hardware*, precisa no minimo de 9gb de memória RAM.
- O grupo não configurou atempadamente o volume necessário para que a base de dados não seja volátil.

Vantagem de usar minikube:

- Fácil instalação e configuração.
- Exige menos recursos que uma máquina Vagrant.
- Possui uma *dashboard* configurada para gestão dos serviços

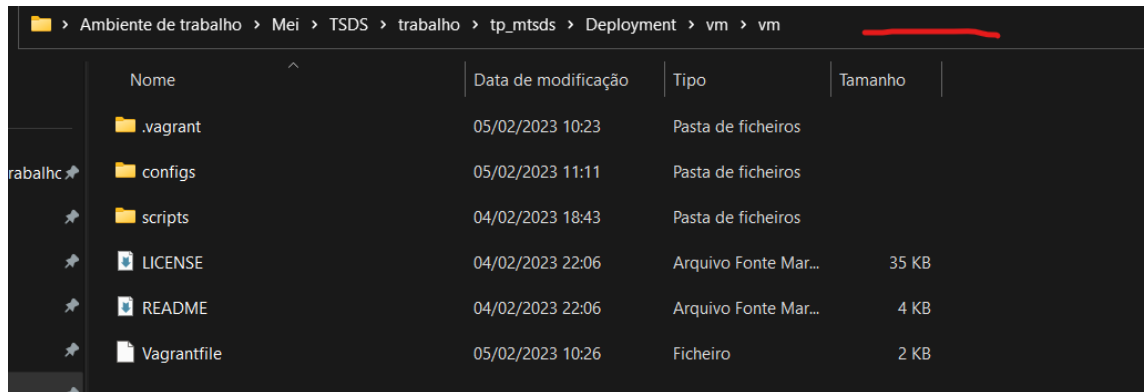
Desvantagem de usar minikube:

- Por vezes o *download* de imagens do docker hub é lento, sendo necessário efetuar o *download* de forma manual.

Instalação da Máquina Vagrant com cluster Kubernetes

Requisitos: Vagrant instalado e Virtual Box.

- 1- Fazer *download* da pasta “Deployment do Repositório”
- 2- Abrir pasta “Deployment/vm/vm”



- 3- Correr comando **vagrant up**. (Demora cerca de 40 minutos a fazer a instalação das 2 vms)
- 4- Correr comando **vagrant ssh master**.
- 5- Correr comando **kubectl get nodes**.

Este é o *output* esperado, foram criadas duas máquinas virtuais na mesma *private network*.

```
vagrant@master-node:~$ kubectl get nodes
NAME             STATUS    ROLES                  AGE     VERSION
master-node      Ready    control-plane         3h34m   v1.25.5
worker-node01    Ready    worker                3h22m   v1.25.5
```

- 6- Correr comando **vim deployment.yaml** para criar um ficheiro Yaml.

```
This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento
vagrant@master-node:~$ vim deployment.yaml
```

- 7- Copiar informação da pasta Deployment /vm/deployment.yaml para dentro do ficheiro criado e guardar as alterações (“qw!”) como mostra as imagens seguintes.

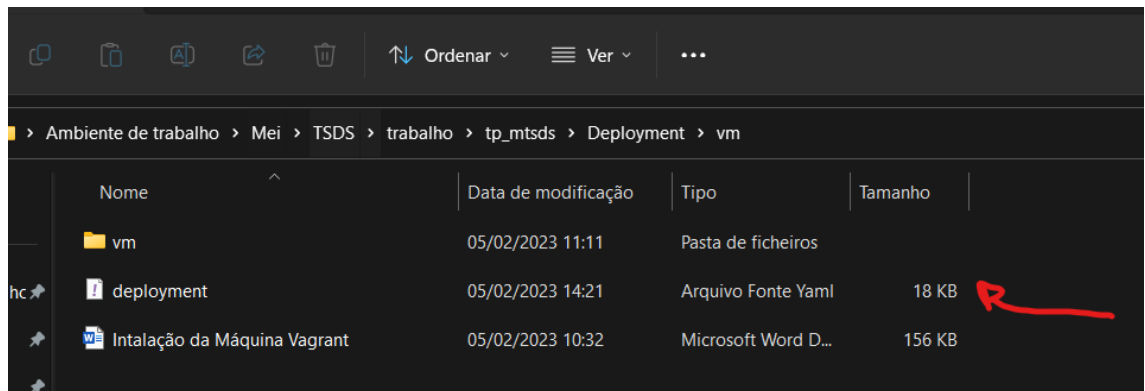


Figure 1 Ficheiro a colocar vm

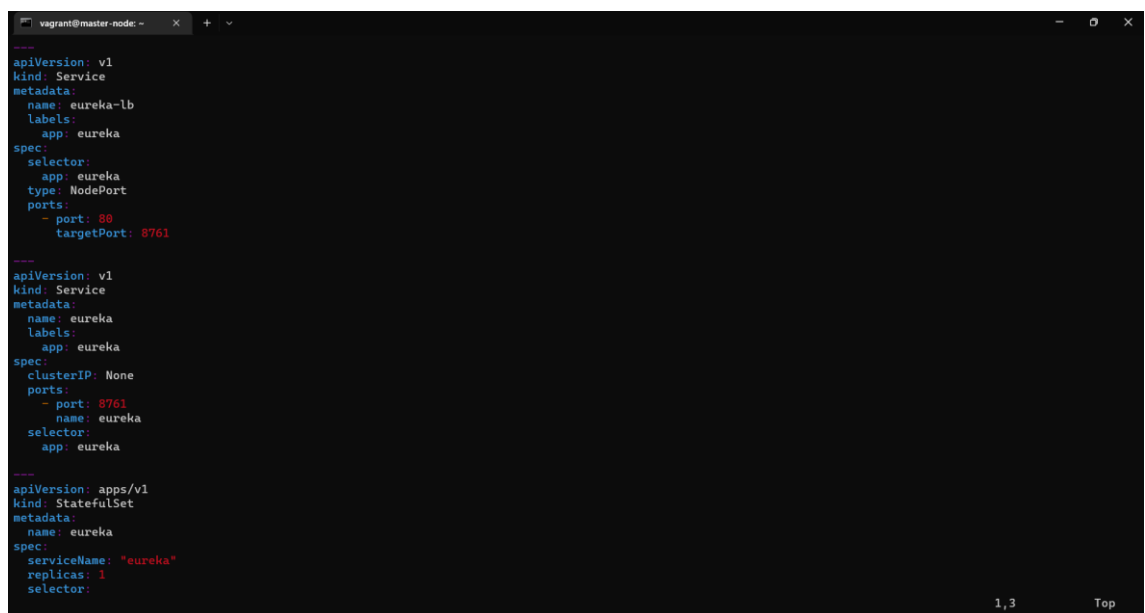
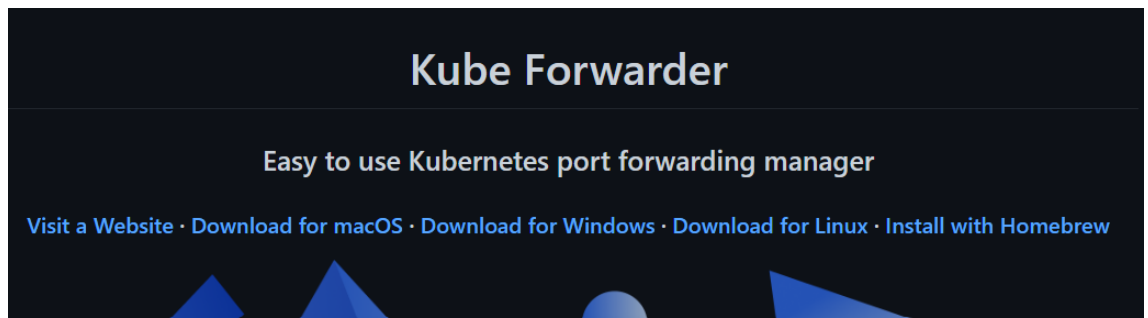


Figure 2 Ficheiro yaml na vm

- 8- Correr comando **kubectl apply -f deployment.yaml**
- 9- Opcional, correr comando **kubectl get deployments** (verificar se está tudo ok)
- 10- Instalar <https://kube-forwarder.pixelpoint.io/>

Nota: Caso o sistema operativo seja Linux ou IOS o *download* deve ser feito pelo seguinte **link**.

<https://github.com/pixel-point/kube-forwarder>



Nota: Quando foi realizada a instalação da VM, foi criada uma pasta chamada configs (Verificar).

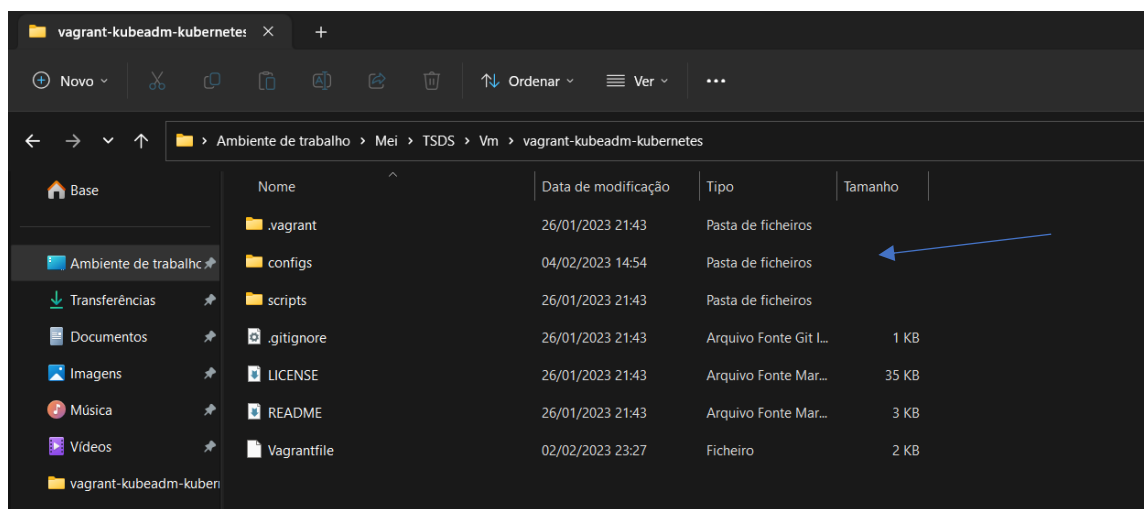
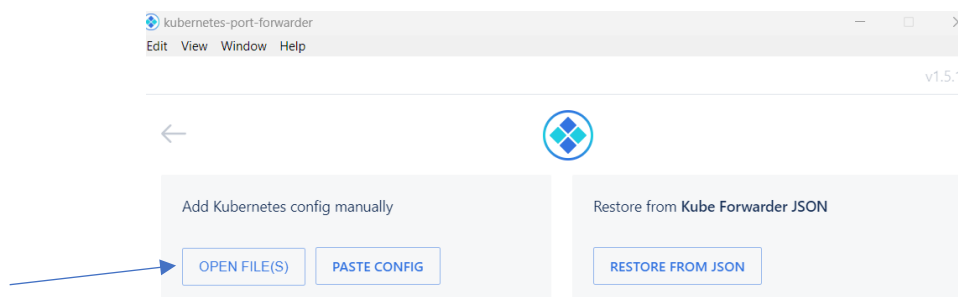
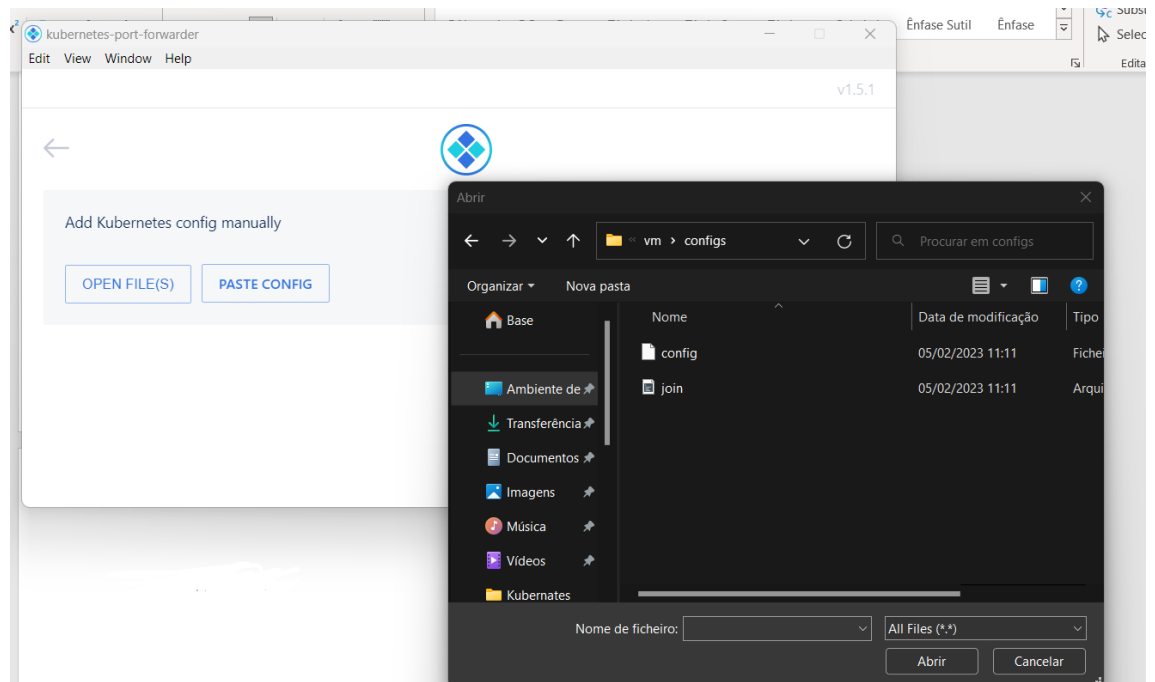


Figure 3 Pasta com configuração da vms

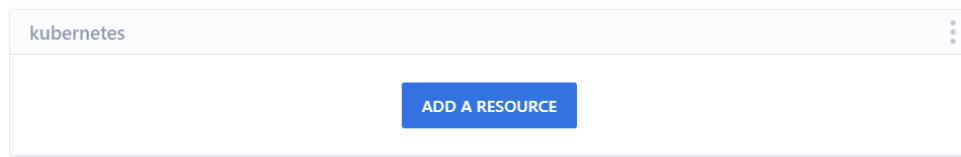
11- Essa pasta config tem as configurações necessárias para adicionar o cluster criado ao KubeForwarder, logo para registar o cluster deve-se clicar em “Open File” e abrir o ficheiro *config*.





12- O último passo é a exportação da porta,

Adicionar novo recurso.



Colocar as seguintes configurações:

Cluster Name <input type="text" value="kubernetes"/>	Namespace <input type="text" value="default"/>
Kind <input type="text" value="Service"/>	Name <input type="text" value="gateway-service"/>
Alias <input type="text" value="Optional..."/>	

Exportar a porta 9191 e guardar as configurações.

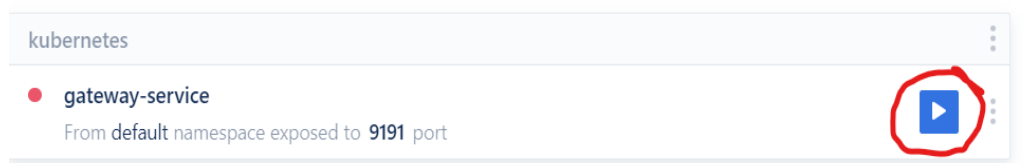
Ports Forwarding

Local Port	Resource Port	
9191	80	✖

CANCEL

SAVE

Correr o recurso criado.



Os serviços estão agora prontos para execução.

Nota: O projeto inclui um *script* que efetua (entre outras coisas) a criação de um utilizador.

Assim, após restabelecer os serviços o projeto está pronto a ser utilizado.

Fazer *request* para autenticação (POST)

<http://localhost:9191/api/auth/login?username=root&password=password>

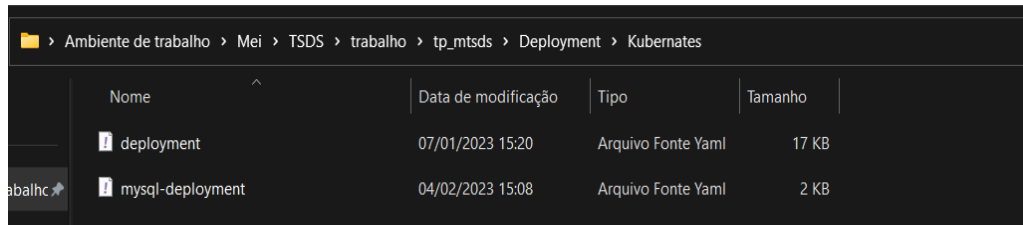
The screenshot shows the Postman interface with a POST request to `http://localhost:9191/api/auth/login?username=root&password=password`. The request body is empty. The response is a JSON object with the following fields:

```
{  "accessToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJyb290IiwiaWVhdHVyZXMlOiSiQURNSU4iLCJ0TElFTlIiOiJlYXNzIiwiaHR0cDovLzE3Mi4xOS40Sjw6OTAwMS9hGkvYXV0aC9sb2dpb1IsImV4cCI6MTY3NTY5NTQ1Nn0.f9sckt6nh8A_HssaE9XeiaTDzUB0KquGBq-xufuUhbY",  "refreshToken": null,  "feature": "ADMIN",  "userId": null}
```


Instalação dos serviços no minikube

Requisitos: Minikube instalado.

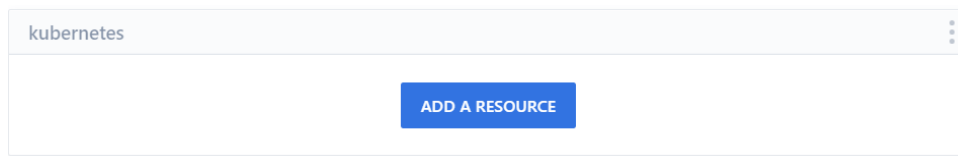
- 1- Correr comando **minikube start** ou **minikube start --driver=hyperv**
- 2- Abrir pasta “Deployment/Kubernetes” através do cmd.



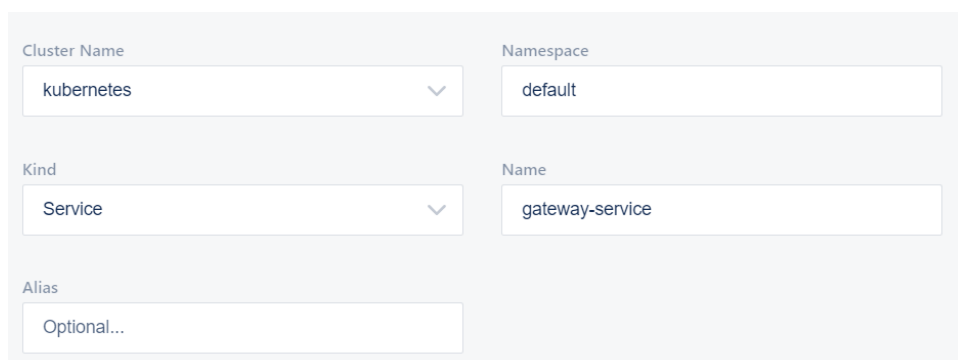
Nome	Data de modificação	Tipo	Tamanho
deployment	07/01/2023 15:20	Arquivo Fonte Yaml	17 KB
mysql-deployment	04/02/2023 15:08	Arquivo Fonte Yaml	2 KB

- 3- Correr comando **kubectl apply -f mysql-deployment.yaml**
- 4- Correr comando **kubectl apply -f deployment.yaml**
- 5- Os serviços serão iniciados.
- 6- O último passo é a exportação da porta

Adicionar novo recurso.

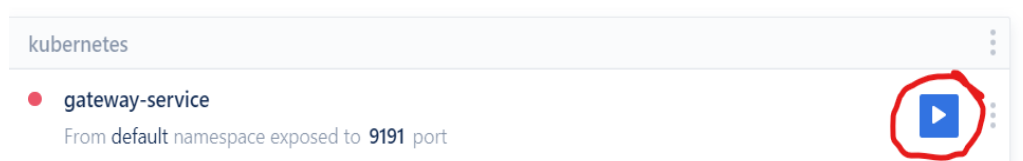


Colocar as seguintes configurações:



Cluster Name	Namespace
kubernetes	default
Kind	Name
Service	gateway-service
Alias	
Optional...	

Correr o recurso criado.



Fazer Request Post

<http://localhost:9191/api/auth/login?username=root&password=password>

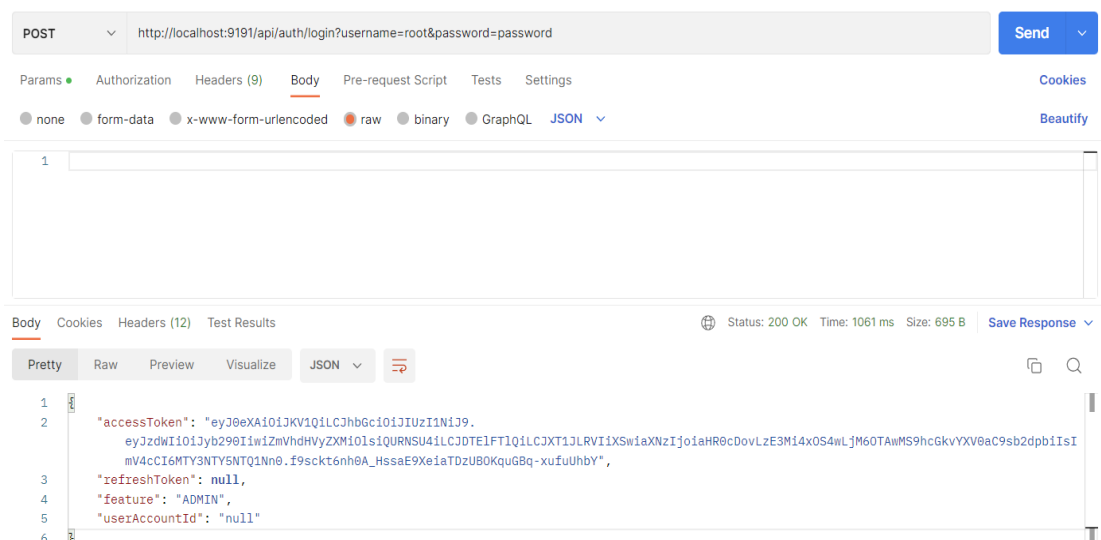


Figure 4 Request Post para login

Nota Importante: Durante o desenvolvimento do projeto o grupo observou algumas dificuldades por parte do minikube em fazer o *download* das imagens docker.

Para ultrapassar isso o grupo executou o comando para *download* manual das imagens, que são apresentadas na seguinte tabela.

Ordem	Comando
1	minikube ssh docker pull hugosilva12/service-registry:0.0.1-SNAPSHOT
2	minikube ssh docker pull hugosilva12/service-gateway:0.0.1-SNAPSHOT
3	minikube ssh docker pull hugosilva12/service-auth:0.0.1-SNAPSHOT
4	minikube ssh docker pull hugosilva12/service-user:0.0.1-SNAPSHOT
5	minikube ssh docker pull hugosilva12/service-purchase:0.0.1-SNAPSHOT
6	minikube ssh docker pull hugosilva12/service-cardisassembly:0.0.1-SNAPSHOT
7	minikube ssh docker pull hugosilva12/service-advertising:0.0.1-SNAPSHOT
8	minikube ssh docker pull hugosilva12/service-sales:0.0.1-SNAPSHOT
9	minikube ssh docker pull hugosilva12/service-precarious:0.0.1-SNAPSHOT
10	minikube ssh docker pull hugosilva12/hystrix:0.0.1-SNAPSHOT

Tabela 1 Tabela com comandos para dwoload de images para o minikube

Instalação do front-end

Nota: O *front-end* não se encontra numa imagem docker, sendo necessário ter o node instalado para o correr.

1. Dentro da pasta Projeto/frontend-service correr comando **npm i;**
2. Correr comando **npm start;**