



## ***Guía de Instalación de Kubernetes, Minikube con Virtual Box y Configuración con Exposición de Servicio en Linux***

Esta guía ha sido desarrollada por Gino Luciano Rojo

- Twitter- recuerdo\_el\_mar
- LinkedIn: <https://www.linkedin.com/in/gino-luciano-rojo/>

Specs utilizados en la Guía :

- Software :
- Kali Linux 20
- Urls de Instalación de Minikube y Kubernetes
- Aplicación desarrollada para el curso con el fin de exponer el servicio

.....ii.....  
.....

Pasos de Instalación :

- Nos posicionamos en el directorio de Descargas o Downloands
- Nos elevamos los privilegios con sudo su

Descargamos desde la siguiente url los instaladores de Kubernetes

Instalar el binario de kubectl con curl en Linux

1. Descargar la última entrega:
2. `curl -LO https://storage.googleapis.com/kubernetes-release/release/\$\(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt\)/bin/linux/amd64/kubectl`

Para descargar una versión específica, reemplaza el comando `$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)` con la versión específica.

Por ejemplo, para descarga la versión v1.21.0 en Linux, teclea:

```
...
curl -LO
https://storage.googleapis.com/kubernetes-release/release/v1.21.0/bin/linux/amd64
/kubectl
...
```

2. Habilita los permisos de ejecución del binario `kubectl`.
3. `chmod +x ./kubectl`
4. Mueve el binario dentro de tu PATH.
5. `sudo mv ./kubectl /usr/local/bin/kubectl`
6. Comprueba que la versión que se ha instalado es la más reciente.

```
kubectl version --client
```



The screenshot shows a terminal window with the following commands and output:

```
root@kali: /home/gino/Downloads
(gino@kali)-[~/Downloads]
$ sudo su
[sudo] password for gino:
(root@kali)-[/home/gino/Downloads]
# curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 45.3M 100 45.3M 0 0 3933k 0 0:00:11 0:00:11 --:--:-- 5026k

(root@kali)-[/home/gino/Downloads]
# chmod +x ./kubectl

(root@kali)-[/home/gino/Downloads]
# sudo mv ./kubectl /usr/local/bin/kubectl

(root@kali)-[/home/gino/Downloads]
# kubectl version --client
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.1", GitCommit:"5e58841cce77d4bc13713ad2b91fa0d961e69192", GitTreeState:"clean", BuildDate:"2021-05-12T14:18:45Z", GoVersion:"go1.16.4", Compiler:"gc", Platform:"linux/amd64"}

(root@kali)-[/home/gino/Downloads]
#
```

También se lo puede instalar de la siguiente manera

```
sudo apt-get update && sudo apt-get install -y apt-transport-https gnupg2 curl
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
```

```
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a
/etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubectl
```

Y en el caso de Windows, se lo puede instalar de la siguiente manera

## Instalar el binario de kubectl con curl en Windows

1. Descargar la última entrega v1.21.0 de [este link](#).

o si tiene curl instalada, utiliza este comando:

```
curl -LO
https://storage.googleapis.com/kubernetes-release/release/v1.21.0/bin/windows/amd64/kubectl.exe
```

Para averiguar la última versión estable (por ejemplo, para secuencias de comandos), echa un vistazo

a <https://storage.googleapis.com/kubernetes-release/release/stable.txt>.

2. Añade el binario a tu PATH.
3. Para asegurar que la versión utilizada sea la más actual, puedes ejecutar:
4. `kubectl version --client`

### Nota:

[Docker Desktop para Windows](#) añade su propia versión de kubectl a PATH.

Si tienes Docker Desktop instalado, es posible que tengas que modificar tu PATH al PATH añadido por Docker Desktop o eliminar la versión de kubectl proporcionada por Docker Desktop.

## Instalar con Powershell desde PSGallery

Si estás en Windows y utilizas el gestor de paquetes [Powershell Gallery](#), puedes instalar y actualizar kubectl con Powershell.

1. Ejecuta los comandos de instalación (asegurándote de especificar una DownloadLocation):
2. `Install-Script -Name 'install-kubectl' -Scope CurrentUser -Force`

3. `install-kubectl.ps1 [-DownloadLocation <path>]`

**Nota:** Si no especificas una `DownloadLocation`, `kubectl` se instalará en el directorio temporal del usuario.

El instalador crea `$HOME/.kube` y crea un archivo de configuración

4. Para asegurar que la versión utilizada sea la más actual puedes probar:

5. `kubectl version --client`

**Nota:** Actualizar la instalación se realiza mediante la re-ejecución de los dos comandos listados en el paso 1.

Instalar en Windows usando Chocolatey o scoop

1. Para instalar `kubectl` en Windows puedes usar el gestor de paquetes [Chocolatey](#) o el instalador de línea de comandos [scoop](#).

o [choco](#)

o [scoop](#)

---

Using [Chocolatey](#).

```
```powershell
choco install kubernetes-cli
```
```

2. Para asegurar que la versión utilizada sea la más actual puedes probar:

3. `kubectl version --client`

4. Navega a tu directorio de inicio:

5. *# Si estas usando cmd.exe, ejecuta: `cd %USERPROFILE%`*

6. `cd ~`

7. Crea el directorio `.kube`:

8. `mkdir .kube`

9. Cambia al directorio `.kube` que acabas de crear:

10. `cd .kube`

11. Configura `kubectl` para usar un clúster remoto de Kubernetes:

12. `New-Item config -type file`

**Nota:** Edita el fichero de configuración con un editor de texto de tu elección, como Notepad.

Una vez verificada la instalacion de Kubernetes, que se debe ver asi

```
root@kali: /home/gino/Downloads
File Actions Edit View Help

(gino@kali)-[~/Downloads]
$ sudo su
[sudo] password for gino:
(root@kali)-[/home/gino/Downloads]
# curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 45.3M 100 45.3M 0 0 3933k 0 0:00:11 0:00:11 --:--:-- 5026k

(root@kali)-[/home/gino/Downloads]
# chmod +x ./kubectl

(root@kali)-[/home/gino/Downloads]
# sudo mv ./kubectl /usr/local/bin/kubectl

(root@kali)-[/home/gino/Downloads]
# kubectl version --client
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.1", GitCommit:"5e58841cce77d4bc13713ad2b91fa0d961e69192", GitTreeState:"clean", BuildDate:"2021-05-12T14:18:45Z", GoVersion:"go1.16.4", Compiler:"gc", Platform:"linux/amd64"}

(root@kali)-[/home/gino/Downloads]
#
```

Necesitamos desactivar los roles de Clusters de Kubernetes, para ello ejecutamos lo siguiente y se debe ver asi

```
kubectl delete clusterrolebinding kubernetes-dashboard
```

```
gino@kali: ~  
File Actions Edit View Help  
  
(gino@kali)-[~]  
$ kubectl delete clusterrolebinding kubernetes-dashboard  
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host  
  
(gino@kali)-[~]  
$ minikube start  
minikube v1.20.0 on Debian kali-rolling  
Using the docker driver based on existing profile  
Starting control plane node minikube in cluster minikube  
Pulling base image ...  
Restarting existing docker container for "minikube" ...  
Preparing Kubernetes v1.20.2 on Docker 20.10.6 ...  
Verifying Kubernetes components ...  
  Using image gcr.io/k8s-minikube/storage-provisioner:v5  
  Using image kubernetesui/dashboard:v2.1.0  
  Using image kubernetesui/metrics-scraper:v1.0.4  
Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 2.384509072s  
Restarting the docker service may improve performance.  
Enabled addons: default-storageclass, storage-provisioner, dashboard  
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default  
  
(gino@kali)-[~]  
$ kubectl delete clusterrolebinding kubernetes-dashboard  
clusterrolebinding.rbac.authorization.k8s.io "kubernetes-dashboard" deleted  
  
(gino@kali)-[~]  
$
```

Por el momento lo dejaremos aqui

## Descarga e Instalacion y configuracion de Minikube



# minikube

## Inicio de minikube

minikube es Kubernetes local y se centra en facilitar el aprendizaje y el desarrollo para Kubernetes.

Todo lo que necesita es un contenedor Docker (o similarmente compatible) o un entorno de máquina virtual, y Kubernetes está a un solo comando de distancia: `minikube start`

### *Lo que necesitarás*

- 2 CPU o más
- 2GB de memoria libre
- 20 GB de espacio libre en disco
- conexión a Internet
- Administrador de contenedores o máquinas virtuales, como: [Docker](#) , [Hyperkit](#) , [Hyper-V](#) , [KVM](#) , [Parallels](#) , [Podman](#) , [VirtualBox](#) o [VMWare](#)

### **1**Instalación

Haga clic en los botones que describen su plataforma de destino. Para otras arquitecturas, consulte [la página de lanzamiento](#) para obtener una lista completa de los binarios de minikube.

Sistema operativo

Linux Mac OS Ventanas

Arquitectura

x86-64

Para instalar minikube en **Windows x86-64** usando la **descarga .exe** :

Descargue y ejecute el [instalador de Windows de minikube](#) independiente .

***Si usó una CLI para realizar la instalación, deberá cerrar esa CLI y abrir una nueva antes de continuar.***

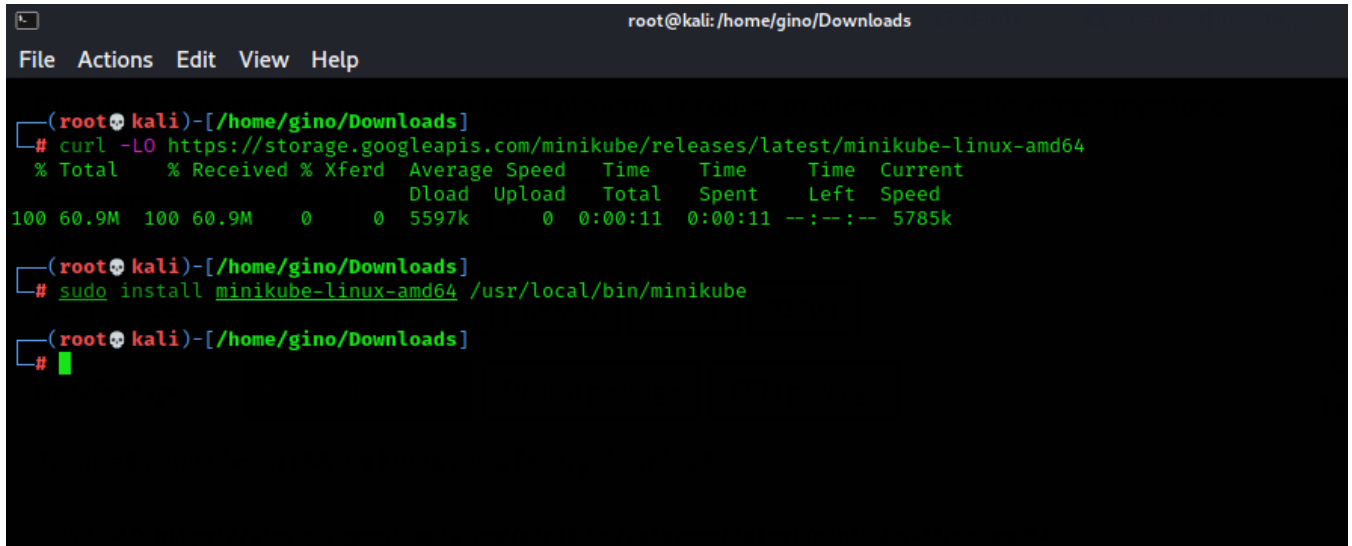
[instalador de Windows de minikube](#)

o

<https://translate.google.com/website?sl=en&tl=es&ajax=1&prev=search&se=1&u=https://storage.googleapis.com/minikube/releases/latest/minikube-installer.exe>

*Para Linux, el cual, será nuestro ejemplo, descargamos con curl, el siguiente link*

```
curl -LO
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

A terminal window titled 'root@kali: /home/gino/Downloads' with a menu bar (File, Actions, Edit, View, Help). The terminal shows three commands being executed: 1. 'curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64' which outputs a progress table. 2. 'sudo install minikube-linux-amd64 /usr/local/bin/minikube'. 3. A prompt '# ' indicating the command is ready to be executed.

| % Total | % Received | % Xferd | Average Speed | Time   | Time    | Time    | Current  |
|---------|------------|---------|---------------|--------|---------|---------|----------|
|         |            |         | Dload         | Upload | Total   | Spent   | Left     |
| 100     | 60.9M      | 100     | 60.9M         | 0      | 0       | 5597k   | 0        |
|         |            |         |               |        | 0:00:11 | 0:00:11 | --:--:-- |
|         |            |         |               |        |         |         | 5785k    |

Debemos instalar Virtualbox y seleccionar con los siguientes comandos, en minikube, la necesidad de usar virtualbox como el driver por defecto, así evitamos que use Docker

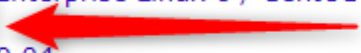
Para instalar Virtualbox en Linux, seguimos el siguiente paso



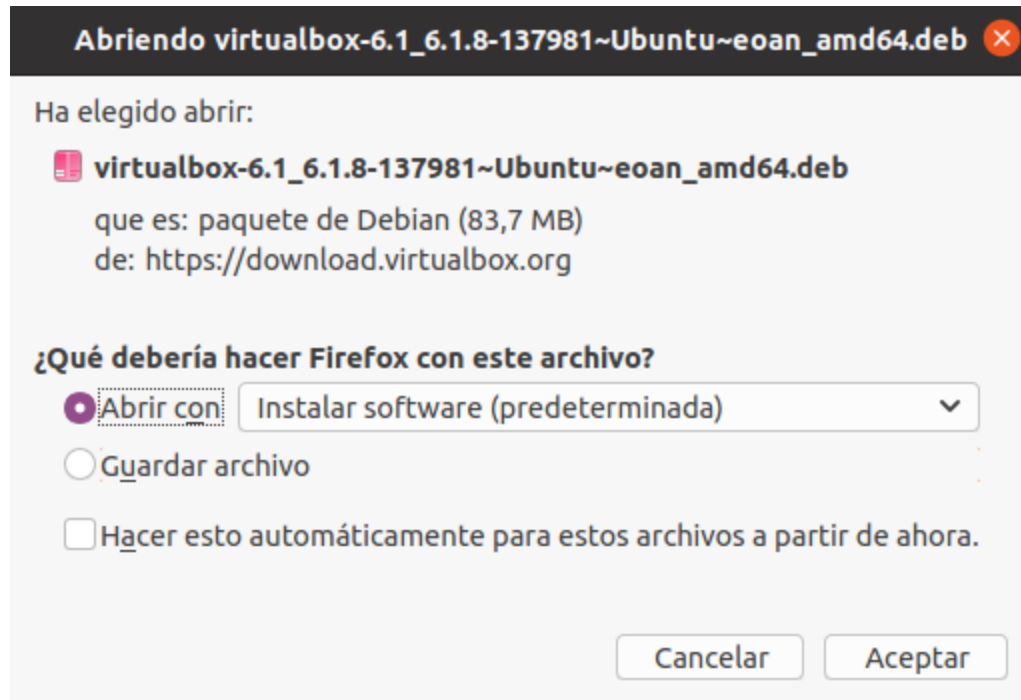
Vamos a <https://www.virtualbox.org/wiki/Downloads>

Y descargamos el .deb desde

### VirtualBox 6.1.8 for Linux

- [Oracle Linux 8 / Red Hat Enterprise Linux 8 / CentOS 8](#)
- [Oracle Linux 7 / Red Hat Enterprise Linux 7 / CentOS 7](#)
- [Oracle Linux 6 / Red Hat Enterprise Linux 6 / CentOS 6](#)
- [Ubuntu 19.10 / 20.04](#) 
- [Ubuntu 18.04 / 18.10 / 19.04](#)
- [Ubuntu 16.04](#)
- [Ubuntu 14.04 / 14.10 / 15.04](#)
- [Debian 10](#)
- [Debian 9](#)
- [Debian 8](#)
- [openSUSE 15.0](#)
- [openSUSE 13.2 / Leap 42](#)
- [Fedora 31](#)
- [Fedora 29 / 30](#)
- [Fedora 26 / 27 / 28](#)
- [All distributions](#) (built on EL6 and therefore not requiring recent system libraries)

Una vez descargado, lo ejecutamos e instalamos VirtualBox



DO NOT COPY

## ***2Inicie su clúster***

Desde una terminal con acceso de administrador (pero que no haya iniciado sesión como root), ejecute:

```
minikube start -driver=virtualbox
```

```
gino@kali: ~  
File Actions Edit View Help  
root@kali: /home/gino x gino@kali: ~ x gino@kali: ~ x  
  
(gino@kali)-[~]  
$ minikube start --driver=virtualbox  
minikube v1.20.0 on Debian kali-rolling  
Using the virtualbox driver based on user configuration  
Downloading VM boot image ...virtualbox driver:  
> minikube-v1.20.0.iso.sha256: 65 B / 65 B [-----] 100.00% ? p/s 0s  
> minikube-v1.20.0.iso: 54.84 MiB / 245.40 MiB 22.35% 3.20 MiB p/s ETA 59s  
minikube start --driver=virtualbox  
  
To make virtualbox the default driver:  
  
minikube config set driver virtualbox  
  
Special features  
  
minikube start supports some VirtualBox specific flags:
```

Se debe ver asi

```
gino@kali: ~  
File Actions Edit View Help  
root@kali: /home/gino x gino@kali: ~ x gino@kali: ~ x  
  
(gino@kali)-[~]  
$ minikube start --driver=virtualbox  
minikube v1.20.0 on Debian kali-rolling  
Using the virtualbox driver based on user configuration  
Downloading VM boot image ...virtualbox driver:  
> minikube-v1.20.0.iso.sha256: 65 B / 65 B [-----] 100.00% ? p/s 0s  
> minikube-v1.20.0.iso: 54.84 MiB / 245.40 MiB 22.35% 3.20 MiB p/s ETA 59s  
minikube start --driver=virtualbox  
  
To make virtualbox the default driver:  
  
minikube config set driver virtualbox  
  
Special features  
  
minikube start supports some VirtualBox specific flags:
```

Podemos verificar la versión de minikube instalado

```
root@kali: /home/gino
File Actions Edit View Help

(root@kali)-[/home/gino]
# wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
--2021-06-05 21:19:10-- https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.172.80, 172.217.172.112, 172.217.172.48, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|172.217.172.80|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 63895278 (61M) [application/octet-stream]
Saving to: 'minikube-linux-amd64'

minikube-linux-amd64 100%[=====] 60.93M 4.01MB/s in 20s
2021-06-05 21:19:30 (3.10 MB/s) - 'minikube-linux-amd64' saved [63895278/63895278]

Step 4: Install kubectl on Ubuntu / Debian
(root@kali)-[/home/gino]
# chmod +x minikube-linux-amd64
# sudo mv minikube-linux-amd64 /usr/local/bin/minikube

(root@kali)-[/home/gino]
# minikube version
minikube version: v1.20.0 https://storage.googleapis.com/kubernetes-release/release/
commit: c61663e942ec43b20e8e70839dcca52e44cd85ae

(root@kali)-[/home/gino]
#
```

Con minikube version

DO NOT COPY

NO DEBE SELECCIONAR EL DRIVER DE DOCKER

```
gino@kali: ~
File Actions Edit View Help

(gino@kali)-[~]
$ minikube start
minikube v1.20.0 on Debian kali-rolling
Automatically selected the docker driver
Starting control plane node minikube in cluster minikube
Pulling base image ...
Downloading Kubernetes v1.20.2 preload ...
> preloaded-images-k8s-v10-v1 ... : 491.71 MiB / 491.71 MiB 100.00% 2.69 MiB
> gcr.io/k8s-minikube/kicbase ... : 358.10 MiB / 358.10 MiB 100.00% 1.86 MiB
> gcr.io/k8s-minikube/kicbase ... : 358.09 MiB / 358.10 MiB 100.00% 3.23 MiB
```

Ahora, iniciamos el dashboard de minikube con

Minikube dashboard

The screenshot shows a terminal window on a Kali Linux system. The terminal output shows the following commands and their results:

```
gino@kali:~$ minikube start
minikube v1.20.0 on Debian kali-rolling
Using the virtualbox driver based on existing profile
Starting control plane node minikube in cluster minikube
Restarting existing virtualbox VM for "minikube" ...

Preparing Kubernetes v1.20.2 on Docker 20.10.6 ...
  • Generating certificates and keys ...
  • Booting up control plane ...
  • Configuring RBAC rules ...
  • Verifying Kubernetes components ...
  • Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: default-storageclass, storage-provisioner
Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default

gino@kali:~$ minikube dashboard --url
Enabling dashboard ...
  • Using image kubernetesui/dashboard:v2.1.0
  • Using image kubernetesui/metrics-scraper:v1.0.4
Verifying dashboard health ...
Launching proxy ...
Verifying proxy health ...
http://127.0.0.1:45621/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/
```

The background shows the Kubernetes Dashboard interface with a sidebar menu and a main content area. The sidebar menu includes options like Overview, Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, and Services. The main content area displays a table of Config Maps.

| Name                 | Namespace | Labels               | Created   |
|----------------------|-----------|----------------------|-----------|
| kubernetes           | default   | component: apiserver | 10.96.0.1 |
| provider: kubernetes | default   | provider: kubernetes | 10.96.0.1 |

Una vez que se ejecuto minikube dashboard y nos brindo la URL, podemos verificar si el status esta correcto con

Minikube status

```
File Actions Edit View Help
Starting control plane node minikube in cluster minikube
Pulling base image ...
Downloading Kubernetes v1.20.2 preload ...
> preloaded-images-k8s-v10-v1 ... : 491.71 MiB / 491.71 MiB 100.00% 2.69 MiB
> gcr.io/k8s-minikube/kicbase ... : 358.10 MiB / 358.10 MiB 100.00% 1.86 MiB
> gcr.io/k8s-minikube/kicbase ... : 358.10 MiB / 358.10 MiB 100.00% 2.17 MiB
Creating docker container (CPUs=2, Memory=2200MB) ...
Preparing Kubernetes v1.20.2 on Docker 20.10.6 ...
  • Generating certificates and keys ...
  • Booting up control plane ...
  • Configuring RBAC rules ...
Verifying Kubernetes components...
  • Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: default-storageclass, storage-provisioner
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

--(gino@kali)-[~]
--$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

Por el momento estamos aquí, ahora, comencemos a iniciar el panel de Kubernetes y acceder a el para ver los nodos, para ello, debemos

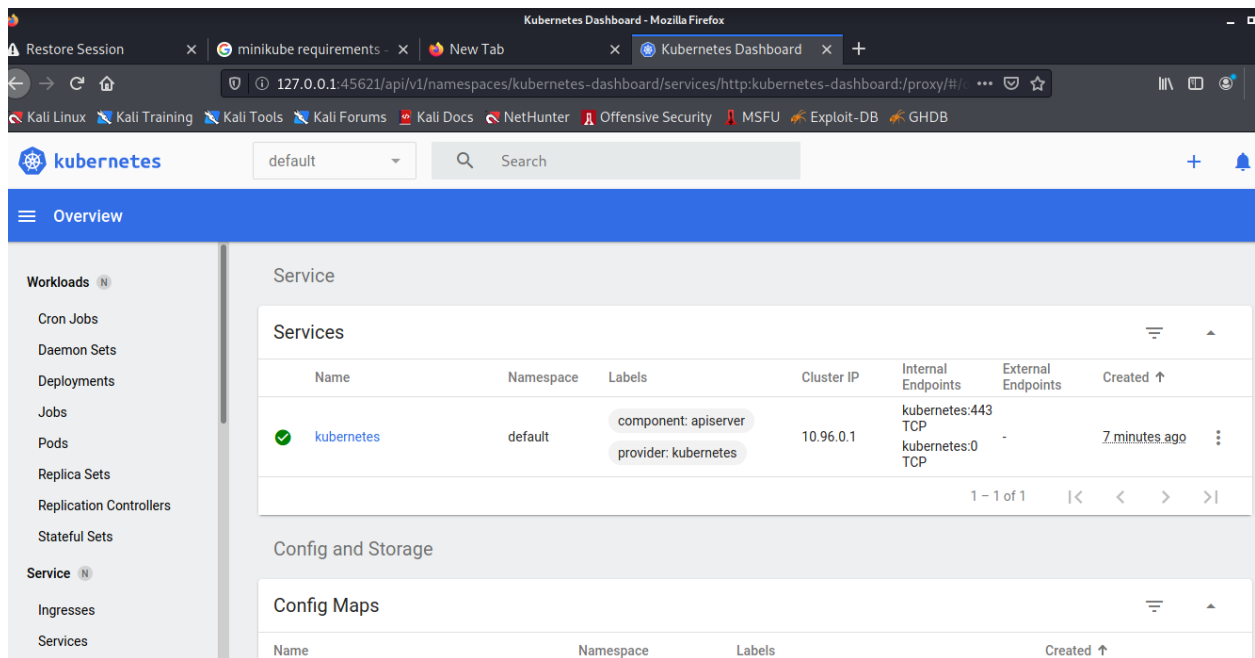
- Si minikube ya se encuentra instalado y ejecutado
- Ejecutamos Kubernetes con los comandos listados debajo y accedemos a la url que nos brinda minikube, es decir, el output de minikube dashboard --url, debemos ingresar a el para ver un panel como el siguiente

Ejecutando minikube dashboard --url

```
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
(gino@kali)-[~]
$ minikube dashboard --url
Enabling dashboard ...
  • Using image kubernetesui/dashboard:v2.1.0
  • Using image kubernetesui/metrics-scraper:v1.0.4
Verifying dashboard health ...
Launching proxy ...
Verifying proxy health ...
http://127.0.0.1:45621/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/

component: apiserver      10.96.0.1      TCP
provider: kubernetes      10.96.0.1      TCP
```

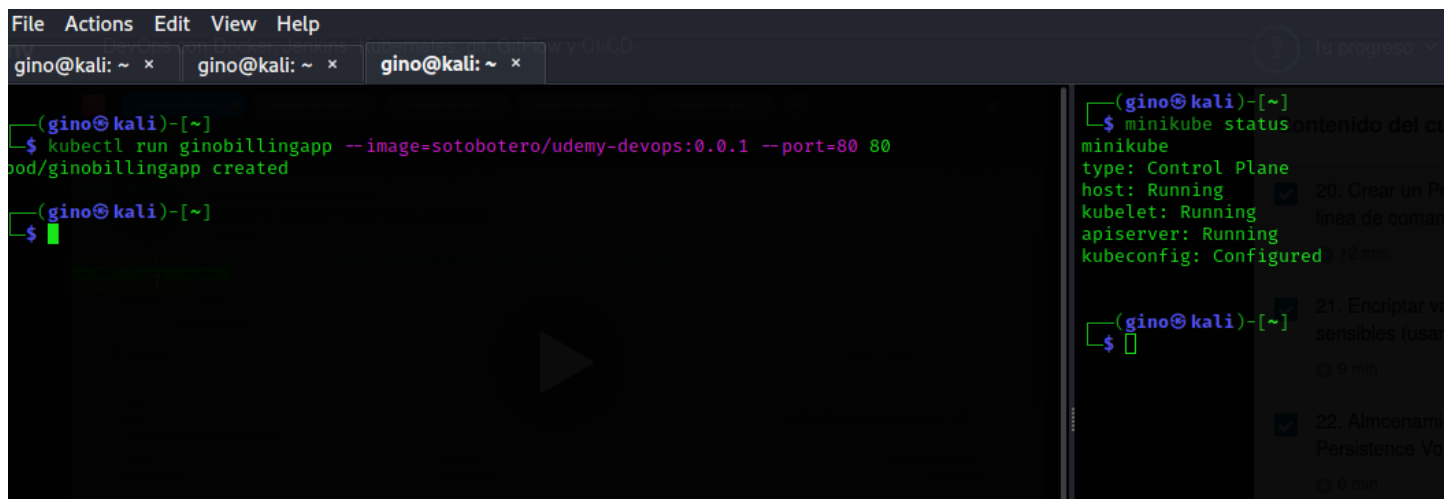
Y se vera así



Aquí, llevaremos adelante una practica simple, iniciaremos un nodo en Kubernetes, se vera como un servicio y lo expondremos para poder acceder a el mediante un navegador, para realizarlo, lo que haremos es lo siguiente

- Creamos un POD en Kubernetes con minikube ejecutando

Kubectrl run elnombredetunodo --image=sotobotero/Udemy-devops:0.0.1 --port=80 80

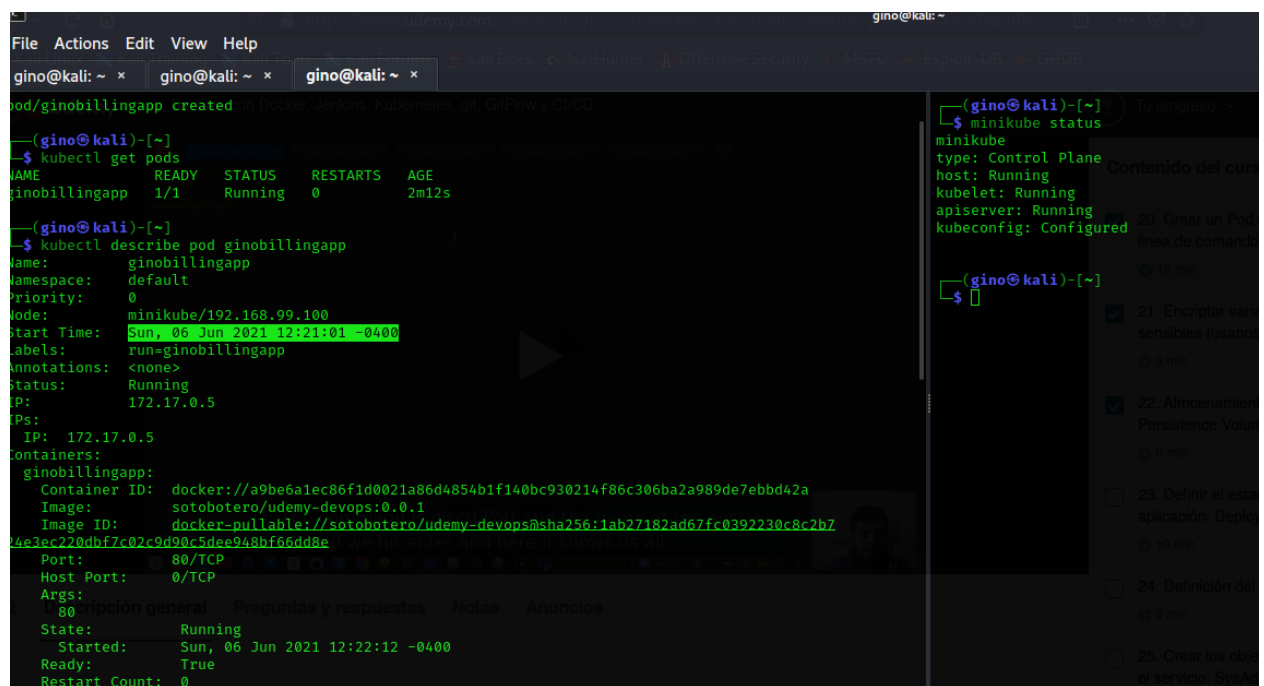


Con el siguiente comando podemos conseguir información de nuestro nodo si , se ejecuto correctamente

Kubectl get pods

Y

Kubectl describe pod elnombredetupod

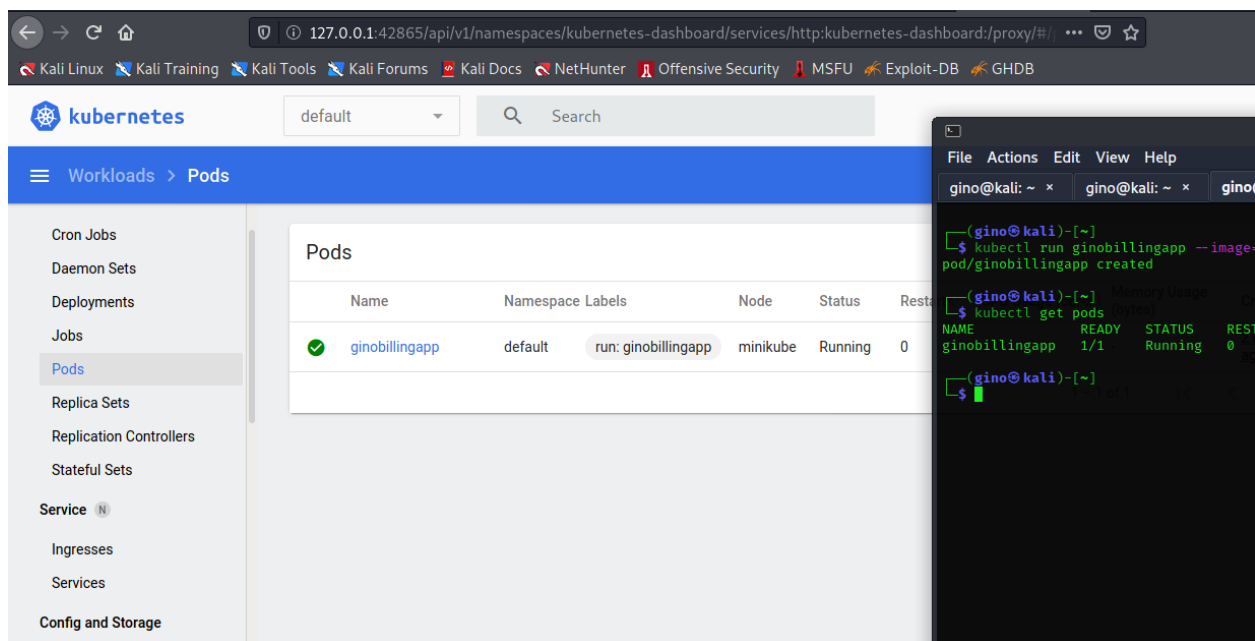


```
gino@kali: ~  
$ kubectl get pods  
NAME                STATUS    RESTARTS   AGE  
ginobillingapp      1/1      Running    0         2m12s  
  
gino@kali: ~  
$ kubectl describe pod ginobillingapp  
Name:                ginobillingapp  
Namespace:           default  
Priority:              0  
Node:                minikube/192.168.99.100  
Start Time:          Sun, 06 Jun 2021 12:21:01 -0400  
Labels:               run=ginobillingapp  
Annotations:          <none>  
Status:              Running  
IP:                  172.17.0.5  
IPs:                  IP: 172.17.0.5  
Containers:  
  ginobillingapp:  
    Container ID:  docker://a9be6a1ec86f1d0021a86d4854b1f140bc930214f86c306ba2a989de7ebbd42a  
    Image:         sotohoto/udemy-devops:0.0.1  
    Image ID:      docker-pullable://sotohoto/udemy-devops@sha256:1ab27182ad67fc0392230c8c2b74e3ec220dhf7c02c9d90c5dee948bf66dd8e  
    Port:          80/TCP  
    Host Port:     0/TCP  
    Args:          /usr/sbin/nginx -g daemon off; --listen=:80 -e /dev/urandom -e /etc/passwd --workers=5  
    State:         Running  
      Started:     Sun, 06 Jun 2021 12:22:12 -0400  
      Ready:       True  
      Restart Count: 0
```

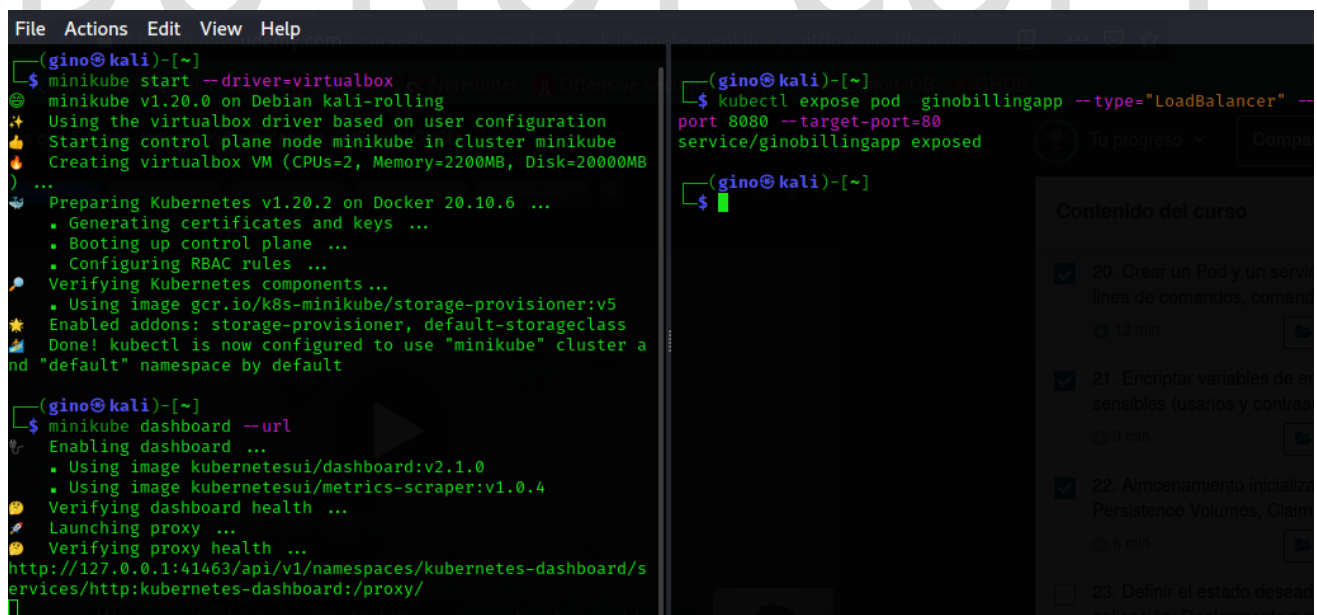
Veremos en el panel de Kubernetes en la web, que el POD se ha creado

También podemos verlo en la terminal ejecutando kubectl get pods





Sin embargo, como vemos no podemos navegar aun dentro de la aplicación que se encuentra dentro del POD, el cual creamos y funciona, pero la dirección IP no nos deja explorarla aun, como permitimos que se explore el pod? Debemos exponerlo, y para ello...

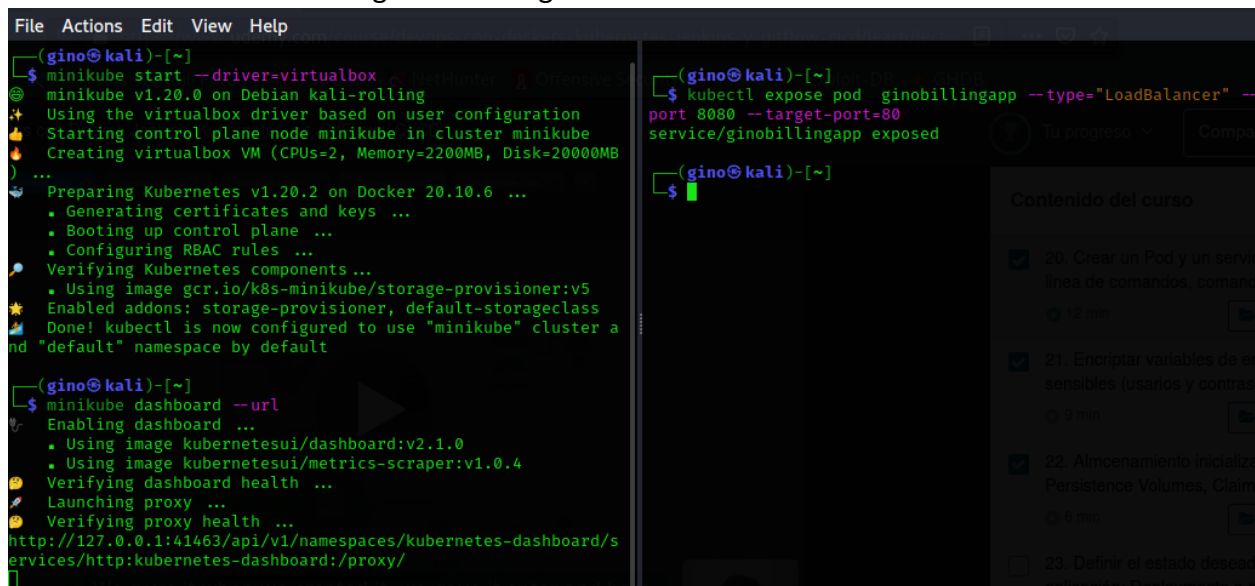


Ejecutando `kubectl expose pod elnombredetuapp -ty pe"LoadBalancer" --port 8080 --target-port=80`

Debería resultar en

Service/elnombredeuapp exposed

Y debería verse tal cual las siguientes imágenes



The image shows two side-by-side screenshots of a terminal window. The left screenshot shows the command `minikube start --driver=virtualbox` being executed, followed by a series of progress messages indicating the setup of the Minikube cluster, including creating a virtualbox VM, preparing Kubernetes, and enabling addons. The right screenshot shows the command `kubectl expose pod ginobillingapp --type="LoadBalancer" --port 8080 --target-port=80` being executed, resulting in the output `service/ginobillingapp exposed`. A sidebar on the right side of the terminal window displays a course content list with items 20, 21, 22, and 23.

```
(gino@kali)-[~]
$ minikube start --driver=virtualbox
minikube v1.20.0 on Debian kali-rolling
Using the virtualbox driver based on user configuration
Starting control plane node minikube in cluster minikube
Creating virtualbox VM (CPUs=2, Memory=2200MB, Disk=20000MB) ...
...
Preparing Kubernetes v1.20.2 on Docker 20.10.6 ...
  Generating certificates and keys ...
  Booting up control plane ...
  Configuring RBAC rules ...
Verifying Kubernetes components...
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

(gino@kali)-[~]
$ minikube dashboard --url
Enabling dashboard ...
  Using image kubernetesui/dashboard:v2.1.0
  Using image kubernetesui/metrics-scraper:v1.0.4
Verifying dashboard health ...
Launching proxy ...
Verifying proxy health ...
http://127.0.0.1:41463/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/

(gino@kali)-[~]
$ kubectl expose pod ginobillingapp --type="LoadBalancer" --port 8080 --target-port=80
service/ginobillingapp exposed

(gino@kali)-[~]
$
```

Confirmamos con `kubectl get services`

```
gino@kali: ~  
File Actions Edit View Help  
- virtualbox | minikube | Kubernetes Dashboard |  
- (gino@kali)-[~]  
$ minikube start --driver=virtualbox  
minikube v1.20.0 on Debian kali-rolling  
Using the virtualbox driver based on user configuration  
Starting control plane node minikube in cluster minikube  
Creating virtualbox VM (CPUs=2, Memory=2200MB, Disk=20000MB)  
...  
Preparing Kubernetes v1.20.2 on Docker 20.10.6 ...  
  Generating certificates and keys ...  
  Booting up control plane ...  
  Configuring RBAC rules ...  
  Verifying Kubernetes components ...  
  Using image gcr.io/k8s-minikube/storage-provisioner:v5  
Enabled addons: storage-provisioner, default-storageclass  
Done! kubectl is now configured to use "minikube" cluster a  
nd "default" namespace by default  
  
(gino@kali)-[~]  
$ minikube dashboard --url  
Enabling dashboard ...  
  Using image kubernetesui/dashboard:v2.1.0  
  Using image kubernetesui/metrics-scraper:v1.0.4  
Verifying dashboard health ...  
Launching proxy ...  
Verifying proxy health ...  
http://127.0.0.1:41463/api/v1/namespaces/kubernetes-dashboard/s  
ervices/http:kubernetes-dashboard:/proxy/  
[~]  
  
(gino@kali)-[~]  
$ kubectl expose pod ginobillingapp --type="LoadBalancer" --  
port 8080 --target-port=80  
service/ginobillingapp exposed  
  
(gino@kali)-[~]  
$ kubectl get services  
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    P  
ORT(S)        AGE  
ginobillingapp LoadBalancer  10.108.101.16  <pending>      8  
080:30461/TCP 74m  
kubernetes    ClusterIP     10.96.0.1      <none>         4  
43/TCP        93m  
  
(gino@kali)-[~]  
$
```

192.168.99.101:30461

Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB

```
gino@kali: ~  
File Actions Edit View Help  
- (gino@kali)-[~]  
$ minikube start --driver=virtualbox  
minikube v1.20.0 on Debian kali-rolling  
Using the virtualbox driver based on user configuration  
Starting control plane node minikube in cluster minikube  
Creating virtualbox VM (CPUs=2, Memory=2200MB, Disk=20000MB)  
...  
Preparing Kubernetes v1.20.2 on Docker 20.10.6 ...  
  Generating certificates and keys ...  
  Booting up control plane ...  
  Configuring RBAC rules ...  
  Verifying Kubernetes components ...  
  Using image gcr.io/k8s-minikube/storage-provisioner:v5  
Enabled addons: storage-provisioner, default-storageclass  
Done! kubectl is now configured to use "minikube" cluster a  
nd "default" namespace by default  
  
(gino@kali)-[~]  
$ minikube dashboard --url  
Enabling dashboard ...  
  Using image kubernetesui/dashboard:v2.1.0  
  Using image kubernetesui/metrics-scraper:v1.0.4  
Verifying dashboard health ...  
Launching proxy ...  
Verifying proxy health ...  
http://127.0.0.1:41463/api/v1/namespaces/kubernetes-dashboard/s  
ervices/http:kubernetes-dashboard:/proxy/  
[~]  
  
service/ginobillingapp exposed  
  
(gino@kali)-[~]  
$ kubectl get services  
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    P  
ORT(S)        AGE  
ginobillingapp LoadBalancer  10.108.101.16  <pending>      8  
080:30461/TCP 74m  
kubernetes    ClusterIP     10.96.0.1      <none>         4  
43/TCP        93m  
  
(gino@kali)-[~]  
$ minikube service ginobillingapp  
+-----+-----+-----+-----+  
| NAMESPACE | NAME       | TARGET PORT | URL                                     |  
+-----+-----+-----+-----+  
| default   | ginobillingapp | 8080        | http://192.168.99.101:30461 |  
+-----+-----+-----+-----+  
Opening service default/ginobillingapp in default browser..  
[~]
```

Ahora, que fue expuesto correctamente el nodo, se ha generado un servicio de ese nodo, y ya podremos visitarlo correctamente, para visitar nuestra aplicación orquestada con Kubernetes y Minikube, debemos conocer la direccion IP del nodo expuesto, nos fijamos el resultado de

Minikube service elnombre de tu app, en el campo URL, y visitamos esa url en el navegador para ver..

Datos de la factura

Gino Luciano Rojo

Para

Educacion

IT

Crear

Listado de facturas

Número Cliente Detalle Monto

gino@kali: ~  
File Actions Edit View Help

gino@kali)~  
\$ minikube start --driver=virtualbox  
minikube v1.20.0 on Debian kali-rolling  
Using the virtualbox driver based on us  
Starting control plane node minikube in  
Creating virtualbox VM (CPUs=2, Memory=...)  
...  
Preparing Kubernetes v1.20.2 on Docker  
• Generating certificates and keys ...  
• Booting up control plane ...  
• Configuring RBAC rules ...  
Verifying Kubernetes components...  
• Using image gcr.io/k8s-minikube/stor...  
Enabled addons: storage-provisioner, de...  
Done! kubectll is now configured to use  
"default" namespace by default

gino@kali)~  
\$ minikube dashboard --url  
Enabling dashboard ...  
• Using image kubernetesui/dashboard:v2...  
• Using image kubernetesui/metrics-scr...  
Verifying dashboard health ...  
Launching proxy ...  
Verifying proxy health ...  
http://127.0.0.1:41463/api/v1/namespaces/ku...  
ervices/http:kubernetes-dashboard:/proxy/  
[]

service/ginobillingapp exposed

gino@kali)~  
\$ kubectl get services

| NAME           | TYPE         | CLUSTER-IP    | EXTERNAL-IP | P |
|----------------|--------------|---------------|-------------|---|
| ginobillingapp | LoadBalancer | 10.108.101.16 | <pending>   | 8 |
| 080:30461/TCP  | 74m          |               |             |   |
| kubernetes     | ClusterIP    | 10.96.0.1     | <none>      | 4 |
| 43/TCP         | 93m          |               |             |   |

gino@kali)~  
\$ minikube service ginobillingapp

| NAMESPACE | NAME           | TARGET PORT | URL                         |
|-----------|----------------|-------------|-----------------------------|
| default   | ginobillingapp | 8080        | http://192.168.99.101:30461 |

Opening service default/ginobillingapp in default browser..

gino@kali)~  
\$

Listo! Ahora nuestra aplicacion funciona!

Confirmamos con

kubectll get services

y para consultar el estado del servicio, ejecutamos

minikube service elnombredetuservicio que fue expuesto.

Al final reiniciamos VirtualBox así si minikube se inicio con Docker, lo cual hará que no funcione

```
[ + ] unattended-upgrades  
[ + ] virtualbox  
[ - ] x11-common  
[ - ] xl2tpd  
  
└─(gino@kali)-[~]  
$ service virtualbox stop  
  
└─(gino@kali)-[~]  
$ service virtualbox start  
  
└─(gino@kali)-[~]  
$ █
```

DO NOT COPY