

Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

## Automatización de despliegues en entornos Cloud

### 1. Objetivos de la actividad

Esta actividad pretende conseguir que te familiarices con las tecnologías de automatización y Cloud.

Implementar una solución IaS Automatizada utilizando Terraform y Ansible

Crear un entorno en Kubernetes

Desplegar una aplicación.

### 2. Introducción

Esta practica consiste en el despliegue de una aplicación utilizando contenedores a través de la tecnología de Kubernetes.

Para el desarrollo de esta actividad se hace uso de tecnologías como Terraform para el despliegue del entorno en Azure, Ansible para la instalación de la infraestructura de kubernetes y configuración del servidor y Kubernetes para el despliegue de una aplicación web.

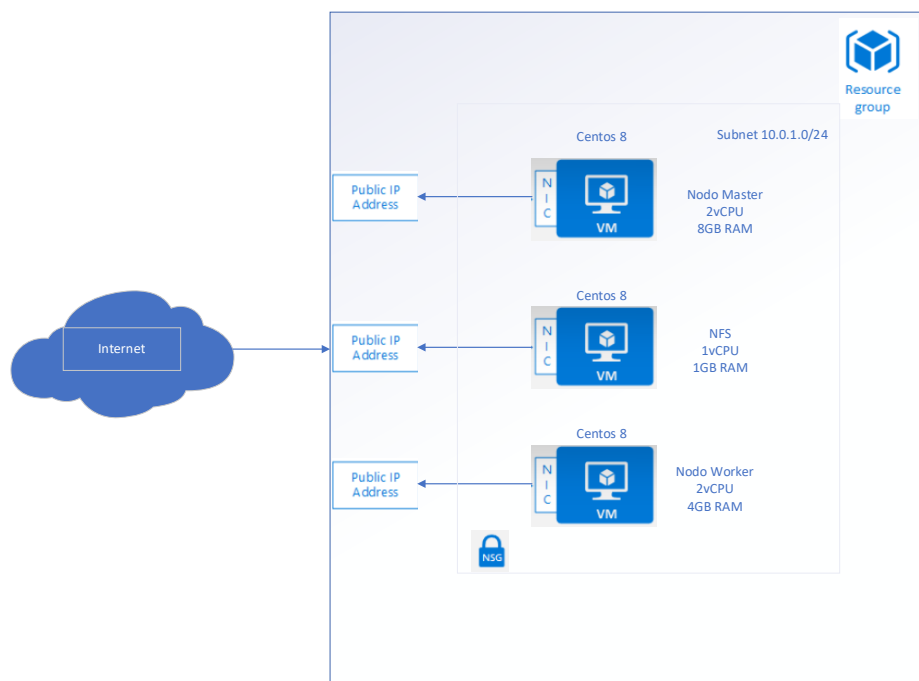
En el desarrollo se utilizó imágenes predefinidas en dockerhub a modo de demostrar que la aplicación funciona.

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps y Cloud	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

### 3. Diagramas:

- **Infraestructura en Azure (Terraform):**

Se desarrollo un programa escrito en HCL usando terraform para desplegar 3 maquinas virtuales que se configuraron como un cluster de kubernetes. Cada VM cuenta con una IP privada y una IP publica para comunicarse con el exterior y se utilizo una imagen gratuita de CENTOS para su sistema operativo. Cada maquina esta asociada a un security group el cual tiene permitido el acceso al puerto 22 para poder acceder vía SSH y al puerto 30000 para que la aplicación sea expuesta hacia el exterior. Toda la instalación se realizo dentro de un resource group que permite agrupar los elementos en cloud.



Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

- **Ansible:**

Se ha utilizado ansible para el despliegue automático de Kubernetes. Los ficheros utilizados para esta práctica están estructurados de la siguiente manera:

**ansible/**

```

├── 01-common.yml
├── 02-nfs.yml
├── 03-master-worker.yml
├── 03-master.yml
├── 04-master.yml
├── 05-workers.yml
├── group_vars
│   ├── app.yml
│   ├── common.yml
│   ├── master_workers.yml
│   ├── master.yml
│   ├── nfs.yml
│   └── workers.yml
├── hosts
├── implementacion-aplicacion.yml
├── implementacion-kubernetes.yml
└── roles
    ├── app
    │   ├── files
    │   │   ├── Paso1_redis-master.yml
    │   │   ├── Paso2_redis-master-service.yml
    │   │   ├── Paso3_redis-slave.yml
    │   │   ├── Paso4_redis-slave-service.yml
    │   │   ├── Paso5_frontend-service.yml
    │   │   └── Paso6_frontend.yml
    │   └── tasks
    │       └── main.yml
    ├── common
    │   └── tasks
    │       └── main.yml
    ├── master
    │   ├── files
    │   └── custom-resources.yml

```

Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

```

| └─ tasks
|   └─ main.yaml
└─ master_workers
  | └─ tasks
  |   └─ main.yaml
  └─ nfs
    | └─ tasks
    |   └─ main.yaml
    └─ workers
      └─ tasks
        └─ main.yaml

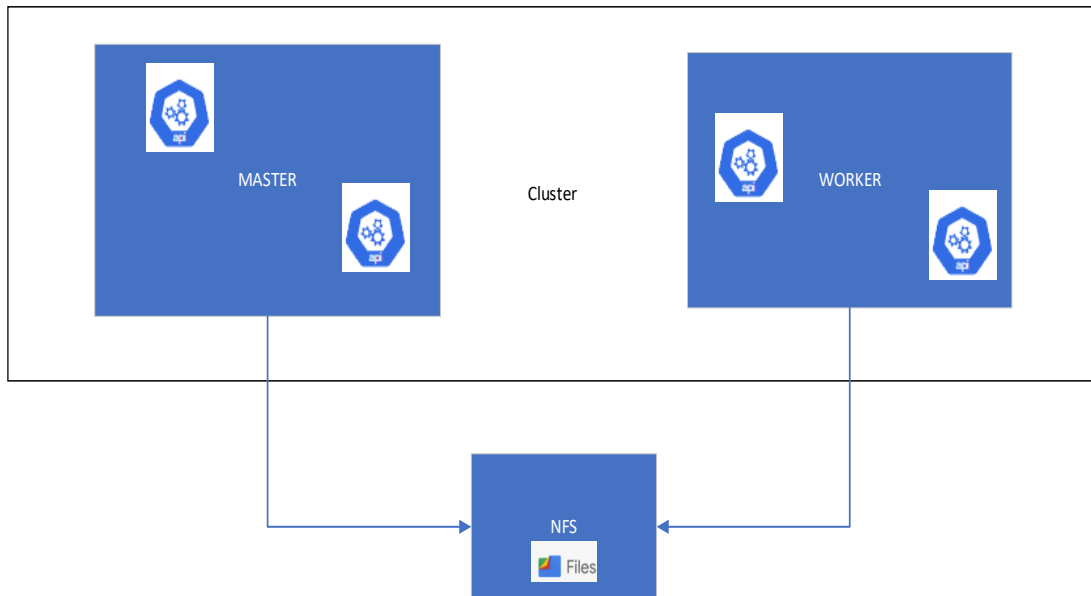
```

Cada archivo yml de la rama principal (ansible) llama a un rol del directorio roles para ejecutar todas las acciones, y a su vez estos roles llaman a las variables que se encuentran en la carpeta group\_vars.

- **Kubernetes:**

Para la instalación y el despliegue de Kubernetes se utilizó Ansible. Esta herramienta nos permite automatizar la instalación y configuración de todo el cluster de kubernetes, además de hacer el despliegue automático de la aplicación. Para este ejercicio se cuenta con un nodo Master y un nodo Worker que serán los encargados de contener toda la infraestructura de kubernetes (pods, servicios, deployments, etc) y un servidor NFS que se utilizara para almacenamiento compartido en la red. Debido a las limitaciones de la cuenta de Azure, solo fue posible desplegar 3 maquinas virtuales y no 4 como sugería el enunciado, debido a que hay limitantes en el número de vCPUs que se pueden configurar.

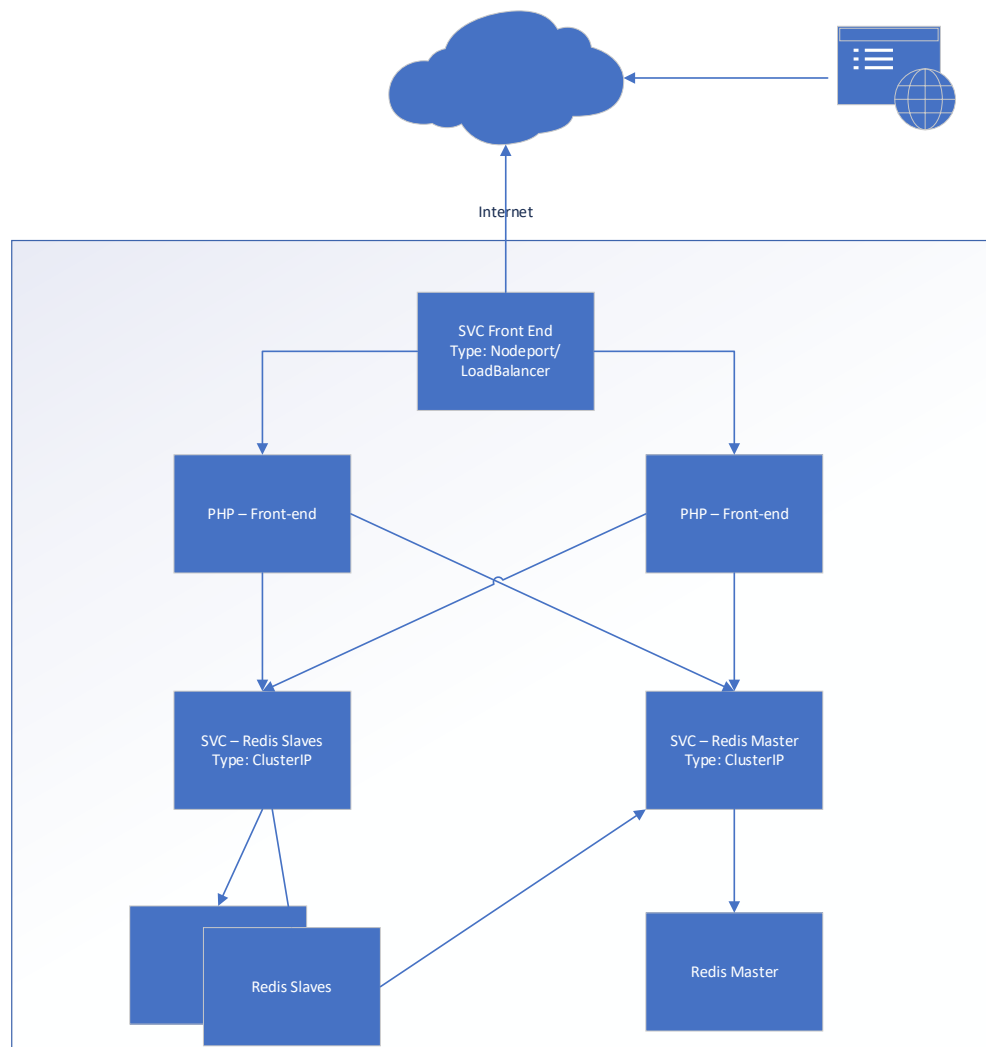
Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	



- **Aplicación:**

La aplicación seleccionada para la práctica es un guestbook, el cual permite a través de una interfaz web almacenar cadenas de texto en una base de datos REDIS. La estructura de la aplicación esta formada por los siguientes elementos:

Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	



En primer lugar, tenemos una Base de Datos Redis Master, la misma que va a contener de manera persistente la información. Por otro lado, tenemos 2 réplicas de nodos esclavos de Redis que funcionan de cache de los datos para mejorar la lectura de los datos. Para comunicar los pods es necesario crear servicios, en este caso al tener mas de 1 replica de los nodos esclavos se creo un servicio de tipo ClusterIP para comunicar a otros estos pods bajo una subred. Por otro lado, se generó otro servicio de tipo CLuster IP (Redis Master) para tener comunicación entre los nodos esclavos y el master. Se generaron 2 réplicas de los front-end con el fin de balancear la carga del

Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

tráfico entrante y finalmente se generó un servicio que puede ser de tipo Nodeport o LoadBalancer para que se comuniquen con el exterior. Para esta práctica y por la facilidad que presenta se usó un servicio del tipo Nodeport que mapea el puerto 80 del contenedor hacia el puerto 30000 del nodo.

#### 4. Descripción de los ficheros

- **Terraform:**

- **Main.tf:** En este fichero configuramos el proveedor de servicio, que en este caso es Azure, agregamos las credenciales, el grupo de recursos que engloba todos los elementos de la infraestructura y el storage account.
- **Networking.tf:** Configuramos la subred, IPs privadas e IPs públicas de las máquinas virtuales a ser desplegadas. En este archivo usamos la subnet 10.0.1.0/24, y hemos definido la IP 10.0.1.10 para el host master, 10.0.1.11 para el host worker y 10.0.1.12 para el host nfs. Las IPs públicas de cada VM se asignan de manera dinámica.
- **Vars.tf:** este archivo contiene todas las variables que van a ser utilizadas en nuestros archivos y que son variables y no estáticas. Aquí tenemos variables como la región que vamos a usar, tipo de la máquina virtual, credenciales de la cuenta azure.
- **Security.tf:** Aquí configuramos las reglas de seguridad, en donde solo habilitamos el puerto 22 para acceder vía ssh y el puerto 30000 para acceder a la aplicación desde afuera.
- **Vm.tf:** configuramos las características de las Máquinas Virtuales, se asigna el tamaño, las NICs, los usuarios y el tipo de disco de cada máquina virtual.

- **Ansible:**

- **01-common.yml:** Este archivo llama al rol common, que se encuentra en el directorio roles y usa las variables que están en el archivo

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps y Cloud	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

“group\_vars/common.yaml”. Al correr este playbook, se instalarán todos los paquetes necesarios para todas las maquinas virtuales, como es el caso de Python, instalamos Chrony para funcionalidades de NTP y brindamos permisos sudo al usuario ansible (usuario con el que vamos a desplegar toda la infraestructura)

- **02-nfs.yml:** Este archivo llama al rol nfs, que se encuentra en el directorio roles y usa las variables que están en el archivo “group\_vars/nfs.yaml”. Al correr este playbook vamos a instalar los paquetes NFS, vamos a montar un directorio compartido y agregar todos los clientes NFS (Workers y Master)
- **03-master-worker.yml:** Este archivo llama al rol master\_worker, que se encuentra en el directorio roles y usa las variables que están en el archivo “group\_vars/master\_worker.yaml”. A través de este playbook vamos a instalar Kubernetes, configurar el firewall para habilitar puertos para kubernetes, montar el NFS, para los servidores Master y Worker.
- **04-master.yml:** Este archivo llama al rol master, que se encuentra en el directorio roles y usa las variables que están en el archivo “group\_vars/master.yaml”. Con este playbook realizaremos tareas específicas sobre el nodo master, como la configuración de kubernetes, instalación del SDN, HAproxy, entre otras.
- **05-workers.yml:** Este archivo llama al rol worker, que se encuentra en el directorio roles y usa las variables que están en el archivo “group\_vars/worker.yaml”. Este playboy se ejecuta sobre los workers y básicamente ejecuta los comandos para unir el nodo al cluster de kubernetes.
- **App:** Este archivo llama al rol app, que se encuentra en el directorio roles y usa las variables que están en el archivo “group\_vars/app.yaml”. Este playbook despliega la aplicación final en el entorno de kubernetes.



Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps y Cloud	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

- **group\_vars:** este directorio contiene todas las variables que van a ser re-utilizadas en los ficheros de ansible.
  - **Roles:** usar ayuda a gestionar de mejor manera las diferentes tareas en ansible. Hemos generado roles con el objetivo de organizar las configuraciones a ser instaladas en cada entorno. Para generar un nuevo rol utilizamos el comando ansible-galaxy
  - **Hosts:** el archivo hosts contiene todas las IPs de cada servidor agrupados, teniendo así un grupo para los “Master”, otro para los “Workers” y otro para los “NFS”. Este file permite identificar en que servidor se va a aplicar las configuraciones.
  - **Implementacion-kubernetes.yml:** dentro de este playbook se ejecutan todos los playbooks descritos anteriormente, pero siguiendo un orden de configuración.
  - **Implementacion-aplicacion.yml:** despliega la aplicación llamando al playbook app.
- **Directorio raíz:**
    - **Deploy\_aplication.sh:** este es un bash script que desplegara automáticamente toda la infraestructura tanto en terraform como en ansible. En primer lugar, se va generar automaticamente un rsa key para poder ser compartido con todas las máquinas virtuales y tener acceso via ssh, después se ejecutará un plan en terrfarom para desplegar toda la infreatructura en Azure. Seguidamente, se generará un archivo hosts con las IPs actualizadas de las maquinas virtuales, mismas que serán automáticamente copiadas al directorio ansible. Finalmente se ejecutará los playbooks de ansible para generar el cluster en kubernetes y se instalará la aplicación.

Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

## 5. Manual de uso de la aplicación:

- **Prerrequisitos:**

- Cuenta en Azure
- Descargar el repositorio:  
[https://github.com/jimpolbmx1/Caso\\_Practico2\\_Juan\\_Pablo\\_davalos.git](https://github.com/jimpolbmx1/Caso_Practico2_Juan_Pablo_davalos.git)
- Maquina con Linux donde se pueda correr un bash script
- En caso de que se despliegue on-premise, configurar 3 máquinas virtuales como describe el ejercicio

- **Como desplegar la aplicación para azure:**

- Clonar el repositorio de github en una maquina local  
[https://github.com/jimpolbmx1/Caso\\_Practico2\\_Juan\\_Pablo\\_davalos.git](https://github.com/jimpolbmx1/Caso_Practico2_Juan_Pablo_davalos.git)
- Ingresar al directorio terraform y abrir con un editor de text el archivo vars.tg
- Modificar los valores de la suscripción con los propios de la cuenta:

Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

```
variable "subscription_id" {
  type      = string
  description = "subscription_id"
  default   = "3038df55-5424-492e-b9b5-eafbdac21ff7"
}

variable "client_id" {
  type      = string
  description = "client_id"
  default   = "21280ce6-5127-4813-a9d9-581db0f2abf3"
}

variable "client_secret" {
  type      = string
  description = "client_secret"
  default   = "9zU8Q~UnzAfT2MKrgRho0NEcNHfYAL001x7pMdgD"
}

variable "tenant_id" {
  type      = string
  description = "tenant_id"
  default   = "899789dc-202f-44b4-8472-a6d40f9eb440"
}
```

- Ya con los valores cambiamos nos movemos al directorio raíz y ejecutamos el script "Deploy\_application.sh"
- Al finalizar, el script nos mostrara la URL de la aplicación, que puede ser ejecutada desde cualquier browser:

```
TASK [app : URL de la aplicacion] *****
ok: [master1] => {
  "msg": "http://20.254.120.242:30000"
}
PLAY RECAP *****
```

- Abrir la URL en el browser en ingresar cualquier cadena de caracteres

Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

← → ↻ ⚠ No es seguro | 20.254.120.242:30000

### Guestbook



- En la parte de abajo se puede verificar como las cadenas se guardan en la base de datos:

← → ↻ ⚠ No es seguro | 20.108.160.154:30000

### Guestbook



test

- Como desplegar la aplicación on-premise (Comprobado con VMWare):
  - Acceder al directorio ansible y abrir el archivo hosts.
  - Editar las IPs de los nodos Master, worker y nfs y agregar los de la infraestructura que ya han sido previamente generadas
  - Por defecto nuestra cluster en kubernetes corre sobre un SDN en flannel, ya que la practica exige que se corra la aplicación sobre Máquinas Virtuales en Azure, pero si se desea correr la aplicación usando el SDN de Calico, es necesario editar las variables que se encuentran en el siguiente directorio: ansible\group\_vars\master.yaml
  - Reemplazar el valor de “flannel” por “calico”

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps y Cloud	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

- Editar el archivo `\group_vars\master_workers.yaml` y remplazar la IP del nodo NFS:

```
private_ip_nfs: "<IP NFS>"
```

- Editar el archivo `\group_vars\nfs.yaml` y remplazar las IPs de los nodos Master y Worker:

```
private_ip_master: "<IP Master>"
```

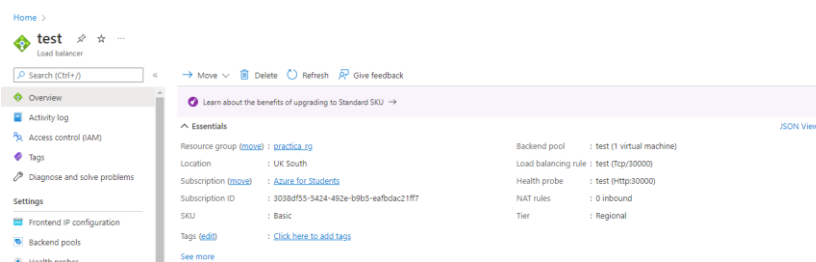
```
private_ip_worker: "<IP Worker>"
```

- Correr los comandos siguientes desde el directorio ansible:
  - `ansible-playbook -i hosts implementacion-kubernetes.yaml`
  - `ansible-playbook -i hosts implementacion-aplicacion.yaml`
- Al correr el ultimo playbook la aplicación mostrara la URL desde donde se puede acceder y probar la aplicación.

## 6. Problemas encontrados en la solución:

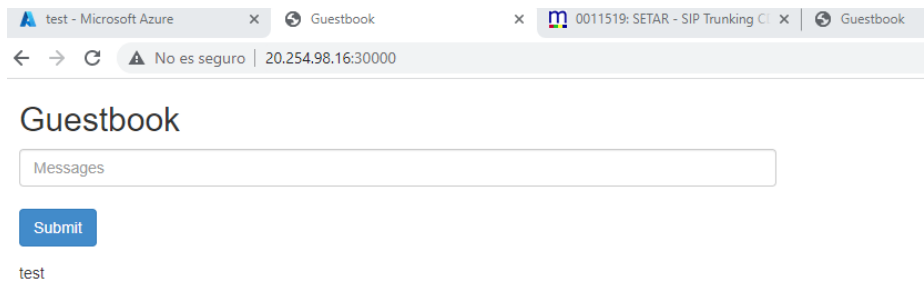
- Para una solución cloud, es recomendable usar un servicio del tipo loadbalancer y tener un balanceador externo. En el caso de Azure se realizaron pruebas generando un loadbalancer externo y se genero un pool interno con las maquinas virtuales con lo cual cuando se hacia una consulta a la IP pública del balanceador de carga apuntaba a cualquier pod de cualquier Nodo. Por ejemplo:

Se genero un loadBalancer denominado test, que apunta al nodo master del cluster en el puerto 30000(donde escucha la aplicacion):



Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	

Al probar logramos el mismo resultado que apuntando a la IP pública del nodo master:



Al estar fuera del alcance de la práctica, solo se probó, pero no se incluyó esta sección dentro del código.

- La cuenta estudiantil de Azure es muy limitada y se tuvo que configurar el nodo master con diferentes características al nodo worker. Cuando se tiene un balanceador de carga en la nube, es necesario que las maquinas virtuales sean de similares características, pero en este caso al no poder generarlas igual, no se pudo probar correctamente el balanceador de carga.

## 7. Conclusiones:

Terraform y ansible son programas que se complementan, con el primero podemos levantar infraestructuras en cualquier sistema de virtualización de manera automatizada (IaC) mientras que ansible nos permite preparar e instalar cualquier tipo de aplicación dentro de la infraestructura. Para esta practica utilizamos terraform para desplegar un entorno en Azure con 3 maquinas virtuales configuradas y por otro lado utilizamos ansible para desplegar un cluster de Kubernetes con una aplicación web.

Asignatura	Datos del alumno	Fecha
<b>Experto Universitario en DevOps y Cloud</b>	Apellidos: Davalos Ríos	22-07-2022
	Nombre: Juan Pablo	