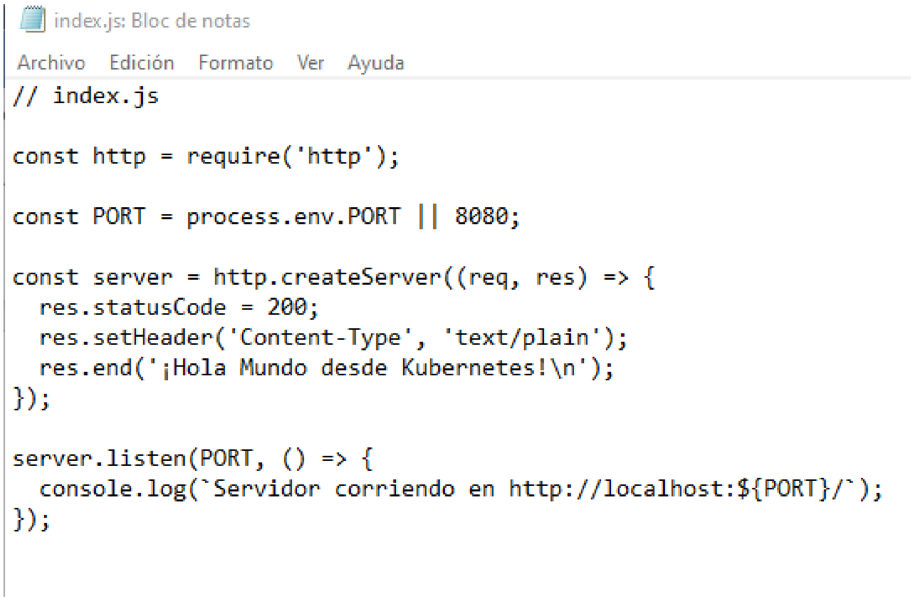
Parcial 3:

Mi disposición de capas funcionó de la siguiente manera:

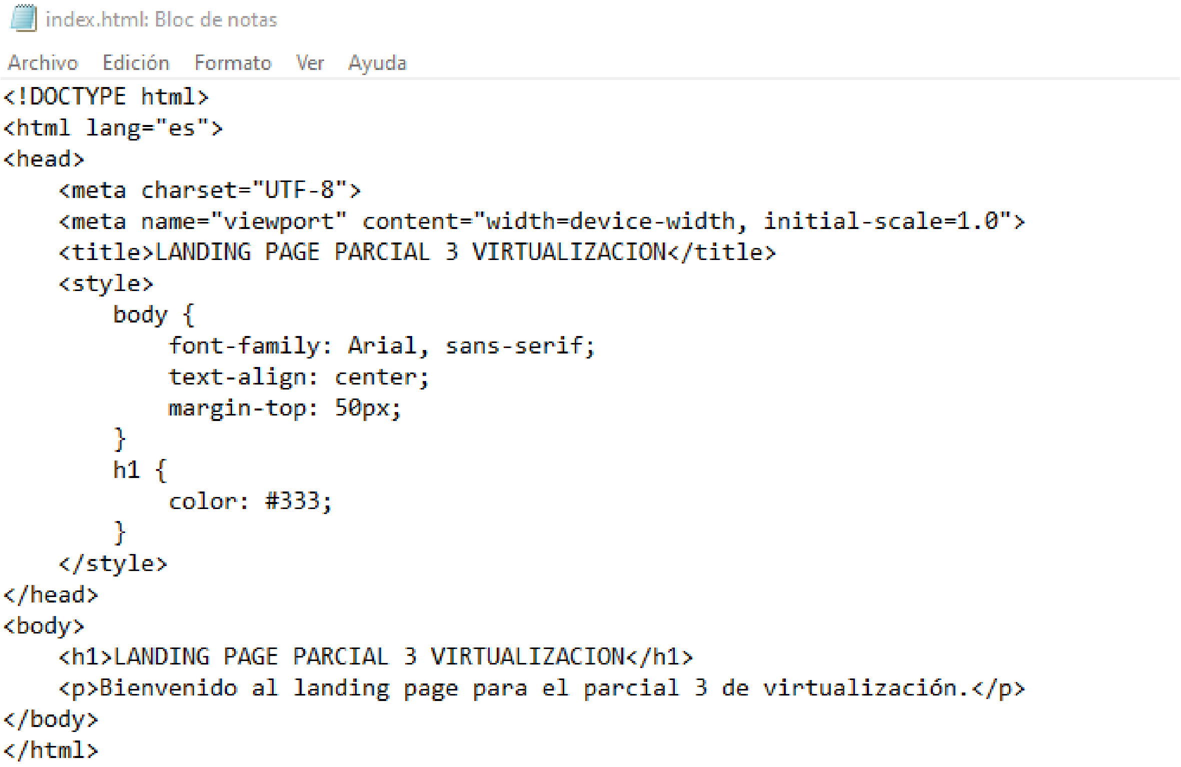
+Api funcional en base a un index.js (Backend)



```
// index.js

const http = require('http');

const PORT = process.env.PORT || 8080;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('¡Hola Mundo desde Kubernetes!\n');
});

server.listen(PORT, () => {
  console.log(`Servidor corriendo en http://localhost:${PORT}/`);
});
```

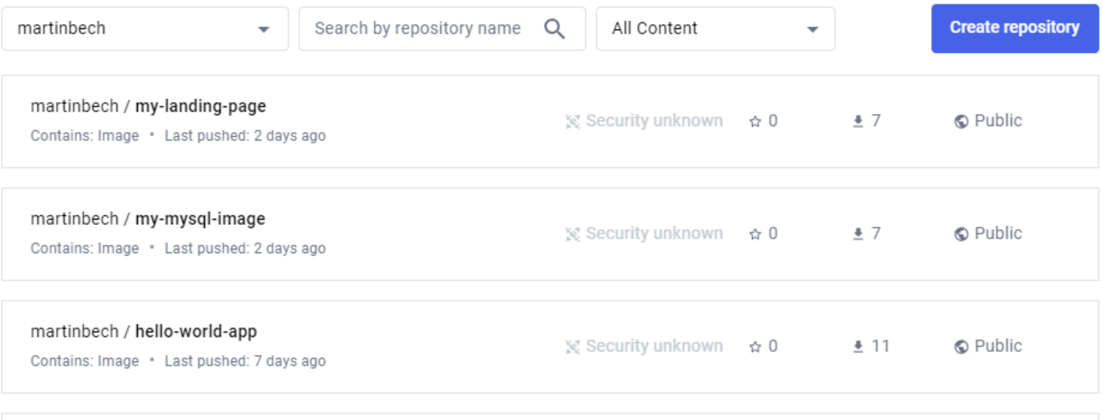
+Base de datos MySql genérica desde Dockerhub (DB)

+Landing Page con texto en nginx (frontend)



```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>LANDING PAGE PARCIAL 3 VIRTUALIZACION</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin-top: 50px;
        }
        h1 {
            color: #333;
        }
    </style>
</head>
<body>
    <h1>LANDING PAGE PARCIAL 3 VIRTUALIZACION</h1>
    <p>Bienvenido al landing page para el parcial 3 de virtualización.</p>
</body>
</html>
```

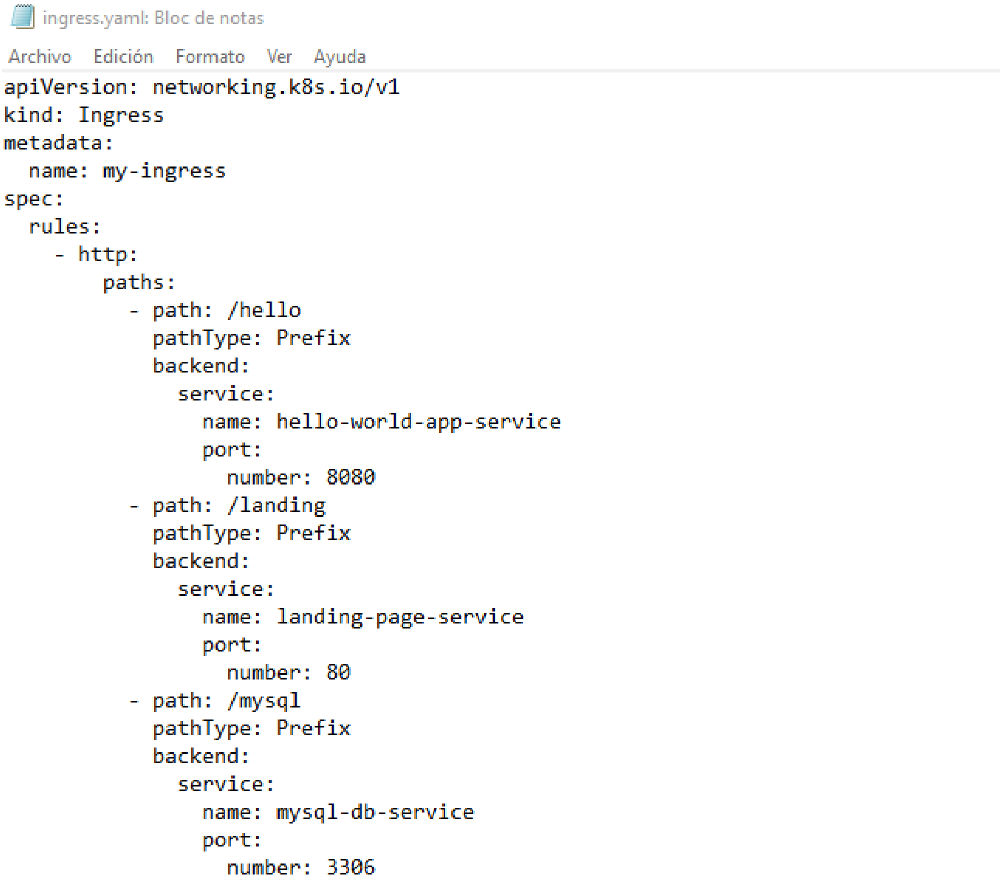
Todas las imágenes fueron montadas en dockerhub



martinbech / **my-landing-page** Contains: Image • Last pushed: 2 days ago	Security unknown	☆ 0	7	Public
martinbech / **my-mysql-image** Contains: Image • Last pushed: 2 days ago	Security unknown	☆ 0	7	Public
martinbech / **hello-world-app** Contains: Image • Last pushed: 7 days ago	Security unknown	☆ 0	11	Public

Posterior, utilicé un Terraform con dos versiones, una donde permite exportar a una ip tanto el api como el landing page y otra donde permite exportar las 3 en el mismo puerto pero con diferente dirección por medio de un ingress yml Y el tuneling de minikube

Ingress:


ingress.yaml: Bloc de notas
Archivo Edición Formato Ver Ayuda

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
    - http:
        paths:
          - path: /hello
            pathType: Prefix
            backend:
              service:
                name: hello-world-app-service
                port:
                  number: 8080
          - path: /landing
            pathType: Prefix
            backend:
              service:
                name: landing-page-service
                port:
                  number: 80
          - path: /mysql
            pathType: Prefix
            backend:
              service:
                name: mysql-db-service
                port:
                  number: 3306
```

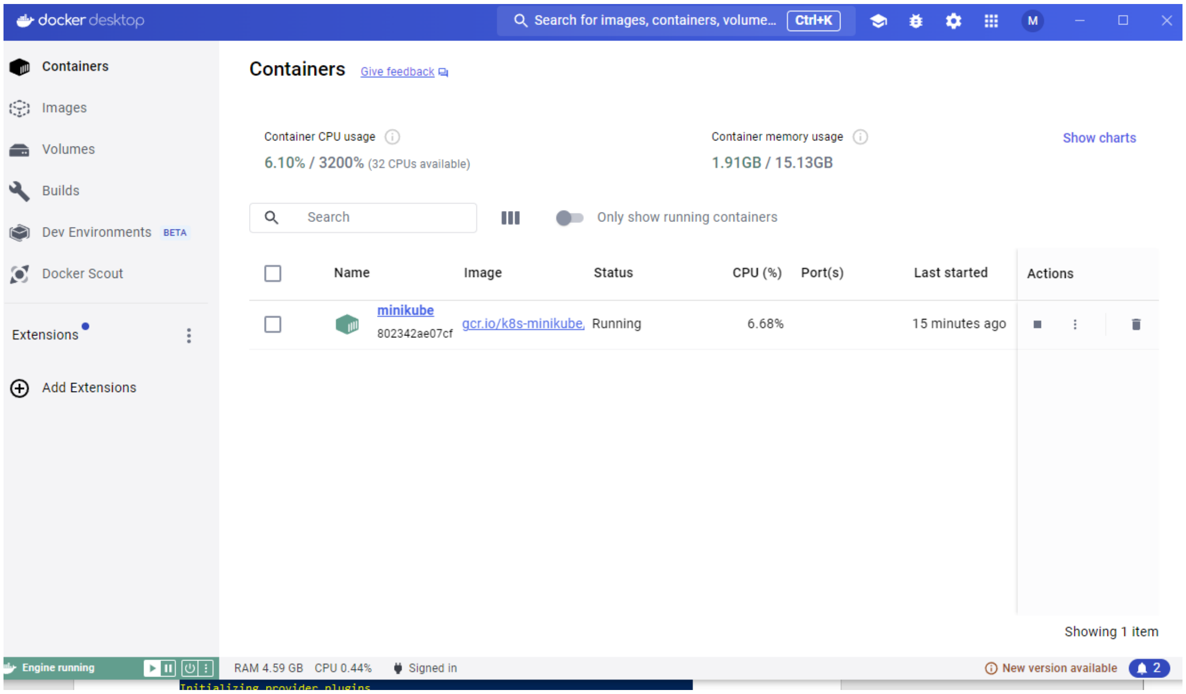
Terraform:

```
provider "kubernetes" {
 config_path    = "~/.kube/config"  # Ruta al archivo de configuración de Kubernetes
 config_context = "minikube"       # Nombre del contexto de Minikube
}

resource "kubernetes_deployment" "landing-page" {
 metadata {
  name = "landing-page"
  labels = {
    app = "landing-page"
  }
}

spec {
  replicas = 1

  selector {
    match_labels = {
      app = "landing-page"
    }
  }

  template {
    metadata {
      labels = {
        app = "landing-page"
      }
    }

    spec {
      container {
        name  = "landing-page"
        image = "martinbech/my-landing-page:latest"
        port {
          container_port = 80
        }
      }
    }
    }
  }
}

resource "kubernetes_service" "landing-page-service" {
  metadata {
    name = "landing-page-service"
  }

  spec {
    selector = {
      app = "landing-page"
    }

    port {
      port        = 80
      target_port = 80
    }
  }
}

resource "kubernetes_deployment" "mysql-db" {
  metadata {
    name = "mysql-db"
    labels = {
      app = "mysql-db"
    }
  }
```

```
}

  spec {
    replicas = 1

    selector {
      match_labels = {
        app = "mysql-db"
      }
    }

    template {
      metadata {
        labels = {
          app = "mysql-db"
        }
      }

      spec {
        container {
          name  = "mysql-db"
          image = "martinbech/my-mysql-image:latest"
          port {
            container_port = 3306
          }
        }
      }
    }
  }
}
```

```
resource "kubernetes_service" "mysql-db-service" {
  metadata {
    name = "mysql-db-service"
  }

  spec {
    selector = {
      app = "mysql-db"
    }

    port {
      port        = 3306
      target_port = 3306
    }
  }
}

resource "kubernetes_deployment" "hello-world-app" {
  metadata {
    name = "hello-world-app"
    labels = {
      app = "hello-world-app"
    }
  }

  spec {
    replicas = 1

    selector {
      match_labels = {
        app = "hello-world-app"
      }
    }

    template {
      metadata {
        labels = {
          app = "hello-world-app"
        }
      }

      spec {
        container {
          name  = "hello-world-app"
          image = "martinbech/hello-world-app:latest"
          port {
            container_port = 8080
          }
        }
      }
    }
  }
}

resource "kubernetes_service" "hello-world-app-service" {
  metadata {
    name = "hello-world-app-service"
  }
  spec {
    selector = {
      app = "hello-world-app"
    }

    port {
      port        = 8080
      target_port = 8080
    }
  }
}
```

Se hizo start al minikube para correr el terraform y que aplicase cambios encima y montase las imágenes



Containers

Container CPU usage
6.10% / 3200% (32 CPUs available)

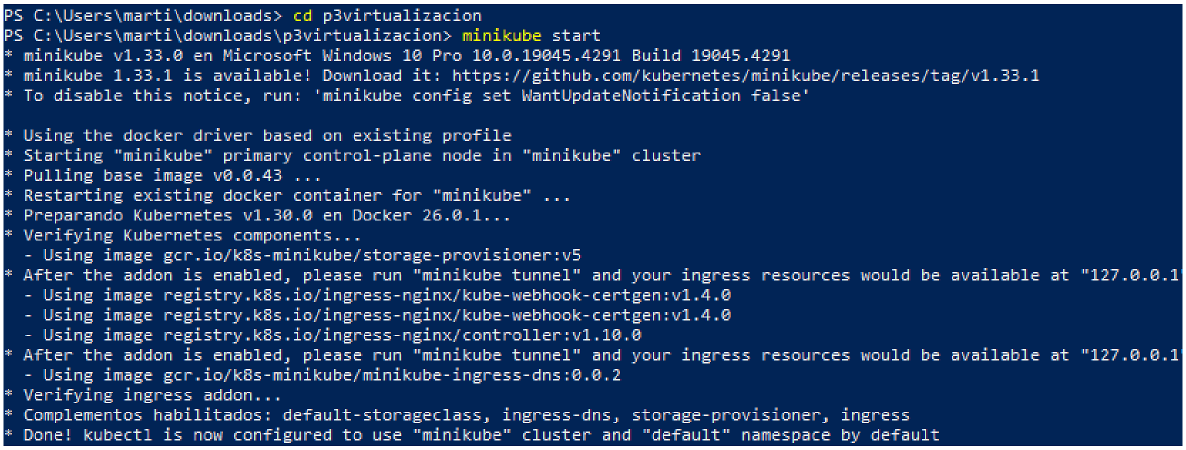
Container memory usage
1.91GB / 15.13GB

□	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
□	minikube 802342ae07cf	gcr.io/k8s-minikube,	Running	6.68%		15 minutes ago	■ ⋮ 🗑

Showing 1 item

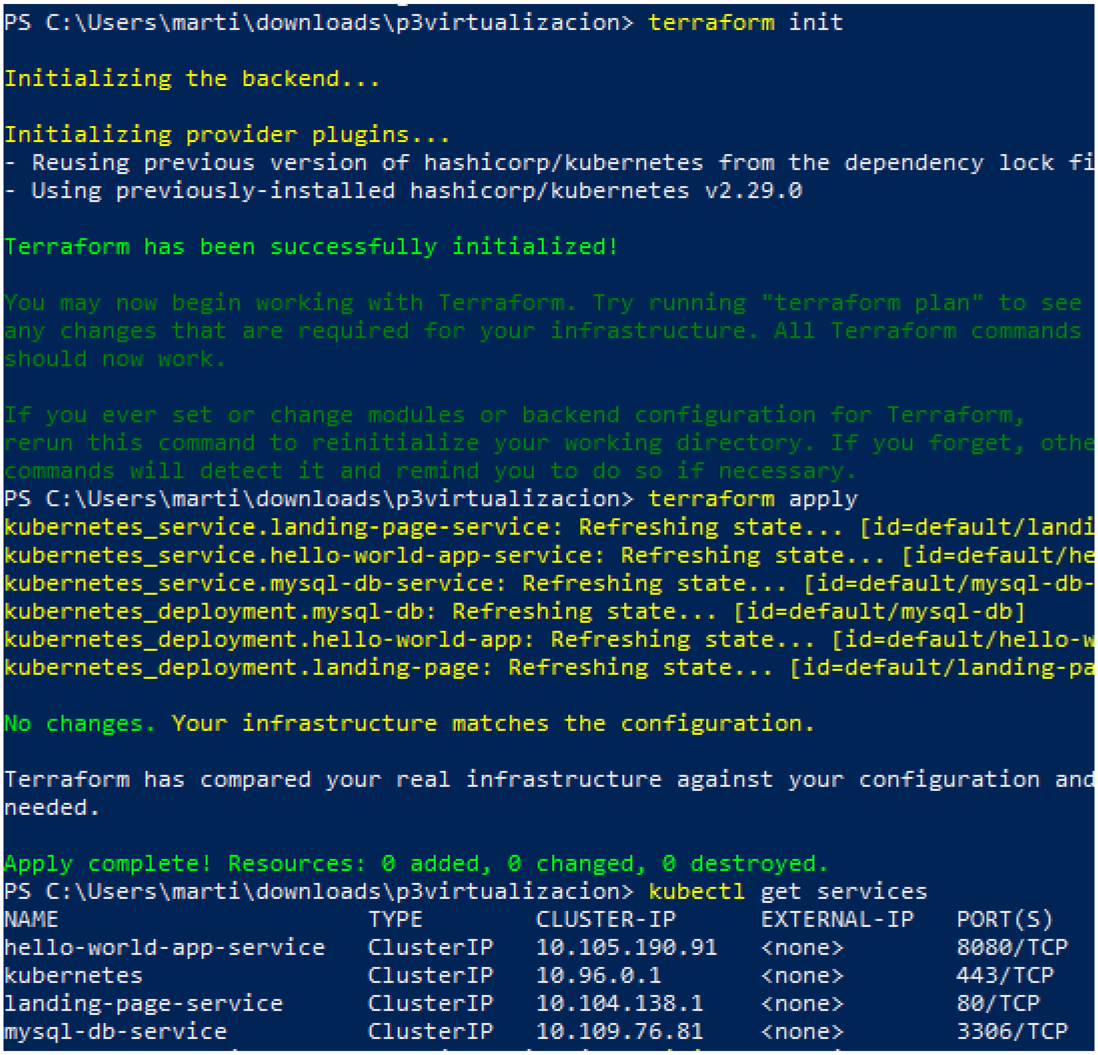
RAM 4.59 GB CPU 0.44%

```
Initializing provider plugins
```



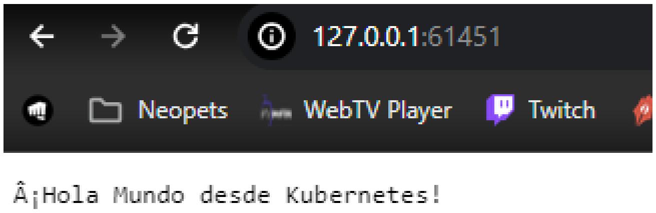
```
PS C:\Users\marti\downloads> cd p3virtualizacion
PS C:\Users\marti\downloads\p3virtualizacion> minikube start
* minikube v1.33.0 en Microsoft Windows 10 Pro 10.0.19045.4291 Build 19045.4291
* minikube 1.33.1 is available! Download it: https://github.com/kubernetes/minikube/releases/tag/v1.33.1
* To disable this notice, run: 'minikube config set WantUpdateNotification false'

* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.43 ...
* Restarting existing docker container for "minikube" ...
* Preparando Kubernetes v1.30.0 en Docker 26.0.1...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* After the addon is enabled, please run "minikube tunnel" and your ingress resources would be available at "127.0.0.1
  - Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.0
  - Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.0
  - Using image registry.k8s.io/ingress-nginx/controller:v1.10.0
* After the addon is enabled, please run "minikube tunnel" and your ingress resources would be available at "127.0.0.1
  - Using image gcr.io/k8s-minikube/minikube-ingress-dns:0.0.2
* Verifying ingress addon...
* Complementos habilitados: default-storageclass, ingress-dns, storage-provisioner, ingress
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Posteriormente se ejecutó el terraform (init y apply) y se verificó que los servicios estuvieran corriendo en el default

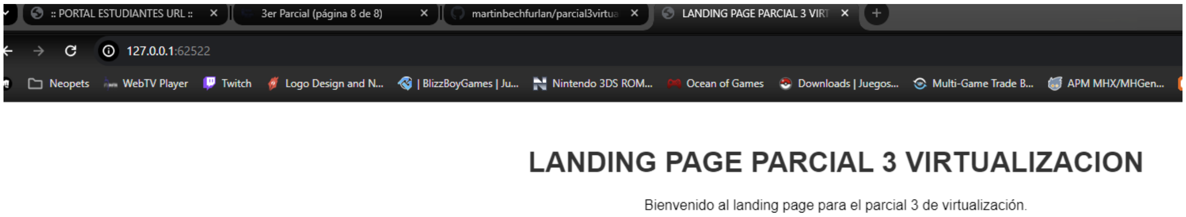


```
PS C:\Users\marti\downloads\p3virtualizacion> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/kubernetes from the dependency lock fi
- Using previously-installed hashicorp/kubernetes v2.29.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, othe
commands will detect it and remind you to do so if necessary.
PS C:\Users\marti\downloads\p3virtualizacion> terraform apply
kubernetes_service.landing-page-service: Refreshing state... [id=default/landi
kubernetes_service.hello-world-app-service: Refreshing state... [id=default/he
kubernetes_service.mysql-db-service: Refreshing state... [id=default/mysql-db-
kubernetes_deployment.mysql-db: Refreshing state... [id=default/mysql-db]
kubernetes_deployment.hello-world-app: Refreshing state... [id=default/hello-w
kubernetes_deployment.landing-page: Refreshing state... [id=default/landing-pa

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and
needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\Users\marti\downloads\p3virtualizacion> kubectl get services
NAME                      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
hello-world-app-service   ClusterIP   10.105.190.91   <none>        8080/TCP
kubernetes                ClusterIP   10.96.0.1       <none>        443/TCP
landing-page-service      ClusterIP   10.104.138.1    <none>        80/TCP
mysql-db-service          ClusterIP   10.109.76.81    <none>        3306/TCP
```

Finalmente se expuso cada uno por separado (dígase el hello world y una landing page para el parcial)



```
Â¡Hola Mundo desde Kubernetes!
```



LANDING PAGE PARCIAL 3 VIRTUALIZACION

Bienvenido al landing page para el parcial 3 de virtualización.