Diego Andres Ramírez Alegria
1139620

Examen Parcial 3

El presente archivo muestra las evidencias de la creación de infraestructura automatizada con código utilizando Terraform localmente.

Los componentes por desplegar son:

1. Frontend (nginx)
2. Backend (API Hola mundo con nodejs)
3. BDD no relacional (Mongo DB)

Dichos componentes están contenerizados y serán desplegados en conjunto en un clúster de Kubernetes.

Procedimiento

1. Crear archivo de terraform (se incluye en el repositorio como pod.tf)
2. Inicializar Terraform

```
PS C:\Repos\virt-demo\terraform\kubernetes\nginx-scalable> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/kubernetes from the dependency lock file
- Using previously-installed hashicorp/kubernetes v2.30.0

Terraform has been successfully initialized!
```

3. Revisar plan de Terraform

```
PS C:\Repos\virt-demo\terraform\kubernetes\nginx-scalable> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # kubernetes_deployment.hola-mundo will be created
  + resource "kubernetes_deployment" "hola-mundo" {
      + id               = (known after apply)
      + wait_for_rollout = true

      + metadata {
          + generation       = (known after apply)
          + labels           = {
              + "App" = "HolaMundo"
            }
          + name             = "api"
          + namespace        = "default"
          + resource_version = (known after apply)
          + uid              = (known after apply)
        }

      + spec {
          + min_ready_seconds         = 0
          + paused                    = false
          + progress_deadline_seconds = 600
```

Diego Andres Ramírez Alegria
1139620

4. Aplicar plan de Terraform

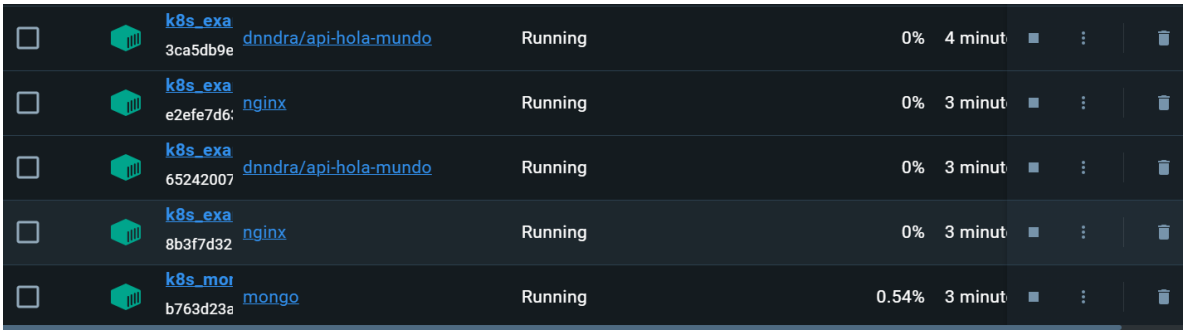
```
kubernetes_deployment.nginx: Creation complete after 36s [id=default/nginx]
kubernetes_deployment.mongo: Creation complete after 36s [id=default/mongo]
kubernetes_deployment.hola-mundo: Creation complete after 36s [id=default/api]
kubernetes_service.hola-mundo: Creating...
kubernetes_service.nginx: Creating...
kubernetes_service.mongo: Creating...
kubernetes_service.mongo: Creation complete after 1s [id=default/mongo]
kubernetes_service.hola-mundo: Creation complete after 1s [id=default/hola-mundo]
kubernetes_service.nginx: Creation complete after 1s [id=default/nginx]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
PS C:\Repos\virt-demo\terraform\kubernetes\nginx-scalable>
```

5. Verificar despliegue en Kubernetes

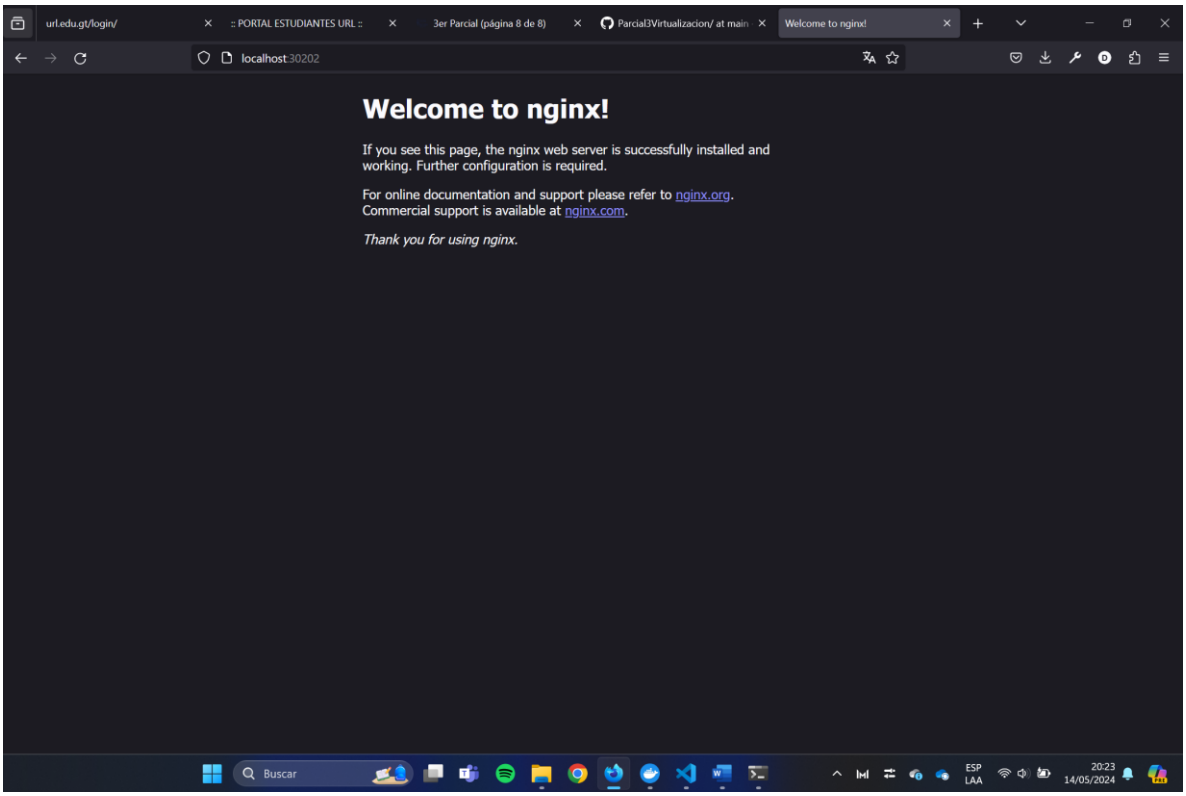
```
PS C:\Repos\virt-demo\terraform\kubernetes\nginx-scalable> kubectl get deployments -o wide
NAME    READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS   IMAGES                          SELECTOR
api     2/2     2            2           2m34s   example      dnndra/api-hola-mundo:latest   App=HolaMundo
mongo   1/1     1            1           2m34s   mongo        mongo:latest                    App=MongoDB
nginx   2/2     2            2           2m34s   example      nginx:latest                    App=ScalableNginx
```

6. Verificando cluster en Docker desktop

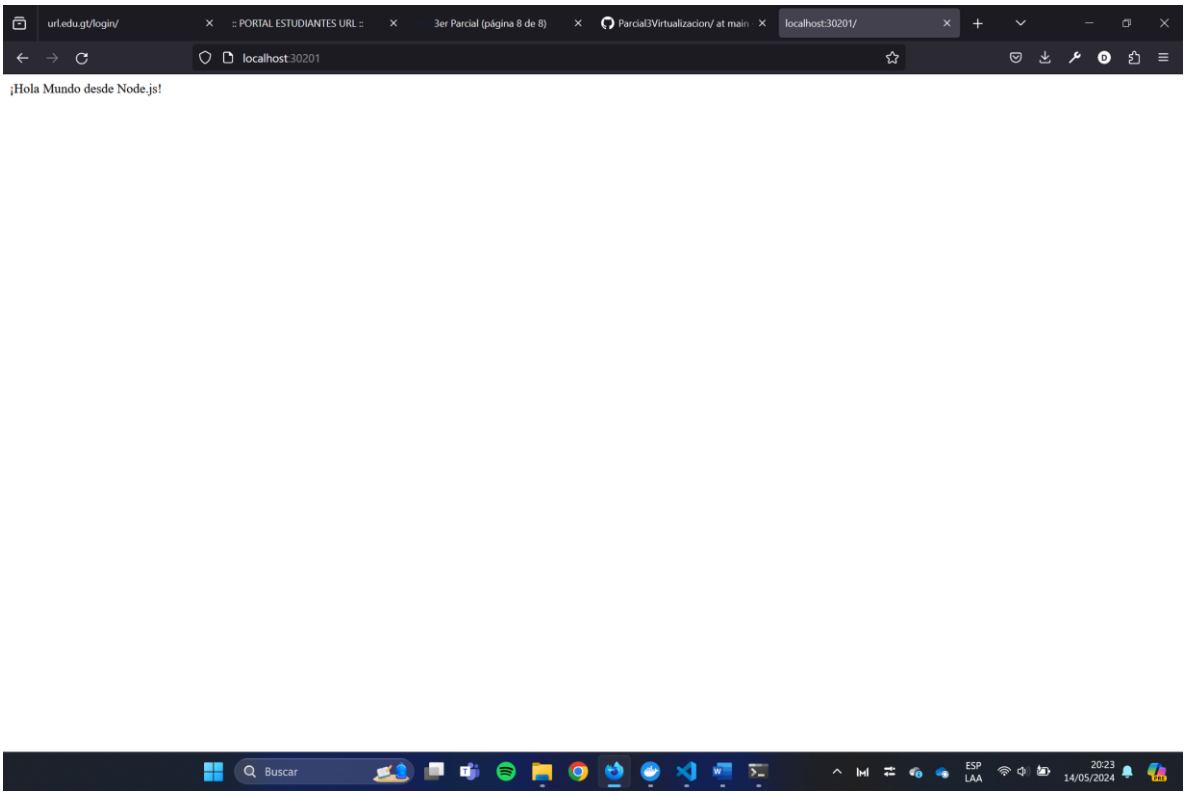


7. Probando Frontend

Diego Andres Ramírez Alegria
1139620

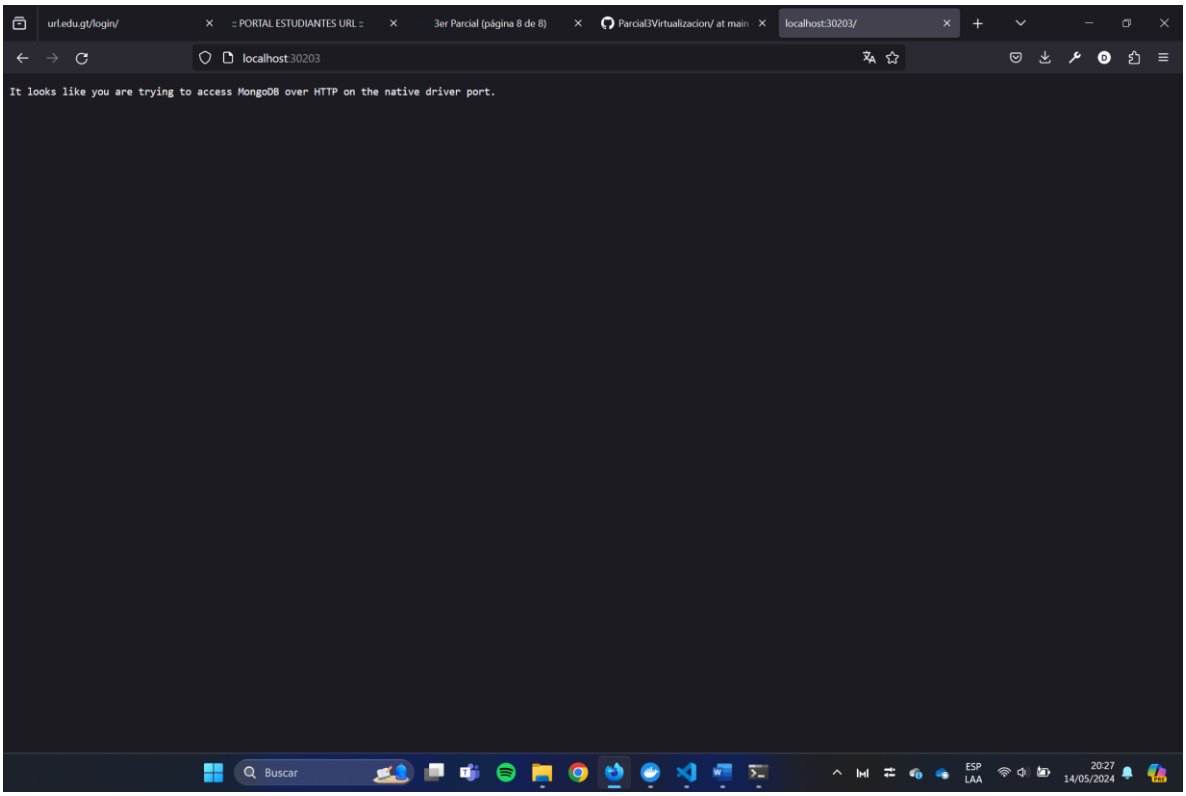


8. Probando Backend



9. Probando Base de datos

Diego Andres Ramírez Alegria
1139620



(nota: accediendo a mongo por http ante la falta de la instalación del cliente de mongo)