ToDo App - Proyecto Final de Containerizacion y Orquestacion

Sistema de gestion de tareas (ToDo List) completamente containerizado y orquestado usando Docker y **Kubernetes (K3D).**

Autor: Wilver Vargas
Tecnologia: Docker Compose, Docker Swarm, Kubernetes (K3D)

REPOSITORIOS

Docker hub

https://hub.docker.com/r/kryshor/todo-backend/tags

Github

https://github.com/W-Varg/ucb_containers_app_todo_list

Tabla de Contenidos

1. Descripcion del Proyecto
2. Arquitectura del Sistema
3. Requisitos Previos
4. Instalacion
5. Despliegue
6. Verificacion y Pruebas
7. Acceso a la Aplicacion

Descripcion del Proyecto

Arquitectura de microservicios con 6 servicios independientes y completamente funcionales:

- **Frontend**: Interfaz web con Nginx y React
- **Backend API**: API REST con Node.js/Express
- **Worker**: Servicio de procesamiento background con Node.js
- **MongoDB**: Base de datos NoSQL
- **Redis**: Cache y cola de mensajes
- **Nginx**: Reverse proxy y load balancer

Caracteristicas Principales

- Containerizacion completa con Docker y Alpine Linux
- Orquestacion con Docker Compose y Docker Swarm
- Despliegue en Kubernetes (K3D)
- Balanceo de carga con Nginx
- Persistencia de datos con MongoDB
- Cache distribuido con Redis
- Procesamiento background con Worker

Servicios Implementados

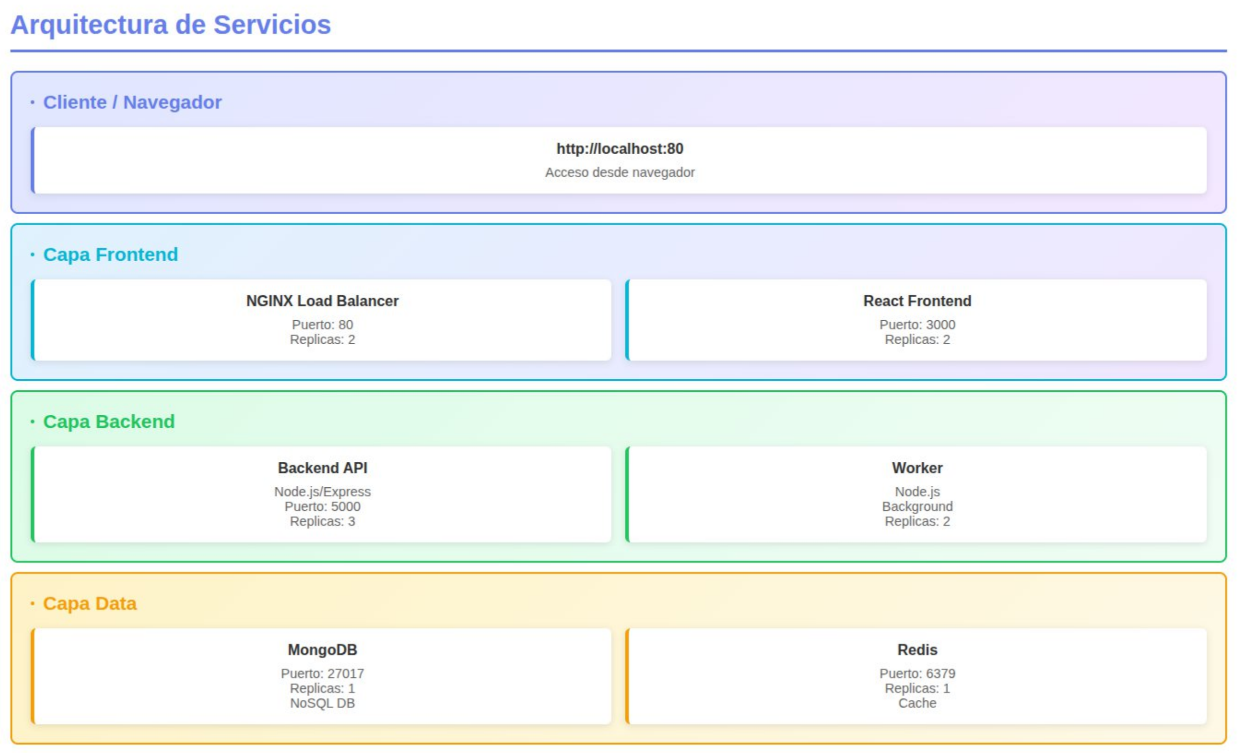
Servicio	Tecnologia	Puerto	Descripcion
Frontend	React 18 + Nginx Alpine	3000	Interfaz de usuario web
Backend	Node.js 18 + Express	5000	API REST para gestion de tareas
MongoDB	MongoDB 7	27017	Base de datos NoSQL
Redis	Redis 7 Alpine	6379	Cache y almacenamiento temporal
Nginx	Nginx Alpine	80	Reverse proxy y load balancer
Worker	Node.js 18	N/A	Procesamiento en background

Arquitectura del Sistema

Componentes por Capa

Capa	Servicio	Tecnologia	Replicas
Load Balancer	Nginx	nginx:alpine	2
Frontend	Web UI	nginx:alpine	2
Backend	REST API	node:18-alpine	3
Worker	Background	node:18-alpine	2
Database	MongoDB	mongo:7-jammy	1
Cache	Redis	redis:7-	1

alpine



Requisitos Previos

Software Necesario

- **Docker** (version 20.10+)
- **Docker Compose** (2.0+)
- **Git**
- **kubectl** (para Kubernetes)
- **K3D** (para Kubernetes)

Verificacion de Requisitos

```
# Verificar Docker

docker --version # Debe mostrar: Docker version 20.10.0 o superior

# Verificar Docker Compose
```

```
docker compose version # Debe mostrar: Docker Compose version 2.0.0 o superior# Verificar Git

git --version
```

Instalacion

- **Instalar Docker**
- **Instalar K3D**
- **Instalar kubectl**
- **Clonar el Proyecto**

```
# Clonar repositorio
$ git clone https://github.com/W-Varg/ucb_containers_app_todo_list.git

cd ucb_containers_app_todo_list
```

Estructura del Proyecto

```
ucb-final/
├── backend/              # API Node.js
│   ├── server.js
│   ├── package.json
│   ├── Dockerfile
│   └── .dockerignore
│
├── frontend/             # Aplicacion React
│   ├── public/
│   │   └── index.html
│   ├── src/
│   │   ├── App.js
│   │   ├── App.css
│   │   ├── index.js
│   │   └── index.css
│   ├── package.json
│   ├── Dockerfile
│   └── .dockerignore
│
├── worker/                    # Servicio de procesamiento
│   ├── worker.js
│   ├── package.json
│   ├── Dockerfile
│   └── .dockerignore
│
├── nginx/                     # Reverse proxy
│   ├── nginx.conf
│   ├── Dockerfile
│   └── .dockerignore
│
├── mongodb-init/              # Scripts de inicializacion
│   └── init-mongo.js
│
├── kubernetes/                # Manifiestos Kubernetes
│   ├── 00-namespace.yaml
│   ├── 01-secrets-configmap.yaml
│   ├── 02-persistent-volumes.yaml
│   ├── 03-mongodb-deployment.yaml
│   ├── 04-redis-deployment.yaml
│   ├── 05-backend-deployment.yaml
│   ├── 06-worker-deployment.yaml
│   ├── 07-frontend-deployment.yaml
│   ├── 08-nginx-loadbalancer.yaml
│   └── 09-version-2-deployments.yaml
│
├── k3d/                    # Manifiestos K3D
│   ├── cluster-config.yaml
│   ├── 00-namespace.yaml
│   ├── 01-config-secrets.yaml
│   ├── 02-persistent-volumes.yaml
│   ├── 03-mongodb.yaml
│   ├── 04-redis.yaml
│   ├── 05-backend.yaml
│   ├── 06-frontend.yaml
│   ├── 07-worker.yaml
│   ├── 08-nginx-ingress.yaml
│   ├── 09-nginx-config.yaml
│   ├── deploy-k3d.sh
│   ├── test-k3d.sh
│   ├── verify-k3d.sh
│   ├── cleanup-k3d.sh
│   └── README-K3D.md
│
├── swarm/                    # Configuracion Docker Swarm
│   ├── stack-deploy.yml
│   ├── stack-simple.yml
│   └── README-SWARM.md
│
├── docker-compose.yml        # Compose para desarrollo local
├── README.md                 # Este archivo
├── CHANGELOG-v1.2.0.md       # Historial de cambios
└── .gitignore
```

Despliegue

Usando Docker Compose

para desarrollo local y pruebas rapidas ejecuta los comandos

```
# 1. Construir imagenes
docker compose build

# 2. Iniciar servicios
docker compose up -d

# 3. Verificar estado
docker compose ps
```

```
# 4. Ver logs
docker compose logs -f backend


# 5. Detener servicios
docker compose down


# 6. Limpiar volumenes
docker compose down -v
```

Usando Docker Swarm

Despliegue en modo cluster con replicacion.

```
# 1. Inicializar Swarm
docker swarm init


# 2. Construir imagenes
docker compose build


# 3. Desplegar stack simple
docker stack deploy -c swarm/stack-simple.yml todoapp


# O desplegar stack completo con versionamiento
docker stack deploy -c swarm/stack-deploy.yml todoapp


# 4. Verificar servicios
docker service ls
```

```
# 5. Ver logs de un servicio

docker service logs todoapp_backend



# 6. Remover stack

docker stack rm todoapp
```

Usando Kubernetes con K3D

Despliegue automatico completo.

```
# 1. Instalar herramientas (solo primera vez)

curl -s https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash

sudo snap install kubectl --classic



# 2. Crear cluster

k3d cluster create --config k3d/cluster-config.yaml



# 3. Cambiar contexto kubectl

kubectl config use-context k3d-todo-cluster



# 4. Construir imagenes

docker compose build



# 5. Importar imagenes al cluster

k3d image import \

    todo-backend:1.2.0 \

    todo-frontend:1.2.0 \

    todo-worker:1.2.0 \

    todo-nginx:1.2.0 \
    -c todo-cluster


# 6. Desplegar aplicacion

chmod +x k3d/deploy-k3d.sh

./k3d/deploy-k3d.sh


# 7. Verificar despliegue

kubectl get all -n todo-app

kubectl get pods -n todo-app

kubectl get services -n todo-app


# 8. Ver logs

kubectl logs -f deployment/backend -n todo-app


# 9. Eliminar cluster

k3d cluster delete todo-cluster
```

Opcion 4: Despliegue Kubernetes Manual

```
# 1. Crear namespace

kubectl apply -f kubernetes/00-namespace.yaml


# 2. Crear secrets y configmaps

kubectl apply -f kubernetes/01-secrets-configmap.yaml


# 3. Crear volumenes persistentes

kubectl apply -f kubernetes/02-persistent-volumes.yaml
```

```
# 4. Desplegar bases de datos

kubectl apply -f kubernetes/03-mongodb-deployment.yaml

kubectl apply -f kubernetes/04-redis-deployment.yaml


# 5. Desplegar aplicacion

kubectl apply -f kubernetes/05-backend-deployment.yaml

kubectl apply -f kubernetes/06-worker-deployment.yaml

kubectl apply -f kubernetes/07-frontend-deployment.yaml


# 6. Configurar load balancer

kubectl apply -f kubernetes/08-nginx-loadbalancer.yaml


# 7. Verificar despliegue

kubectl get all -n todoapp
```

Verificacion y Pruebas

Docker Compose

```
# 1. Crear namespace

kubectl apply -f kubernetes/00-namespace.yaml


# 2. Crear secrets y configmaps

kubectl apply -f kubernetes/01-secrets-configmap.yaml


# 3. Crear volumenes persistentes

kubectl apply -f kubernetes/02-persistent-volumes.yaml
```

```
# 4. Desplegar bases de datos

kubectl apply -f kubernetes/03-mongodb-deployment.yaml

kubectl apply -f kubernetes/04-redis-deployment.yaml


# 5. Desplegar aplicacion

kubectl apply -f kubernetes/05-backend-deployment.yaml

kubectl apply -f kubernetes/06-worker-deployment.yaml

kubectl apply -f kubernetes/07-frontend-deployment.yaml


# 6. Configurar load balancer

kubectl apply -f kubernetes/08-nginx-loadbalancer.yaml


# 7. Verificar despliegue

kubectl get all -n todoapp
```

Docker Swarm

```
# Listar servicios

docker service ls

# Ver estado del servicio

docker service ps todoapp_backend


# Ver logs

docker service logs todoapp_backend -f


# Inspeccionar servicio

docker service inspect todoapp_backend
```

Kubernetes

```
# Ver todos los recursos

kubectl get all -n todo-app

# Ver pods especificos

kubectl get pods -n todo-app

kubectl get pods -n todo-app -w

# Ver logs

kubectl logs -f deployment/backend -n todo-app

kubectl logs POD_NAME -n todo-app
```

Acceso a la Aplicacion

URLs Disponibles

Componente	URL	Puerto
Frontend	http://localhost	80
Backend API	http://localhost:5000	5000
MongoDB	localhost	27017
Redis	localhost	6379

Puntos de Acceso por Entorno

Docker Compose

- Frontend: http://localhost
- Backend: http://localhost:5000
- API Health: http://localhost:5000/health

Docker Swarm

- Frontend: http://localhost
- Backend: http://localhost:5000
- API Health: http://localhost:5000/health

Kubernetes/K3D

- Frontend: http://localhost:9080 (si usa port-forward)
- Backend: http://localhost:9500 (si usa port-forward)

Comandos Utiles

Docker

```
# Construir imagenes con tags especificos

docker build -t kryshor/todo-backend:1.2.0 ./backend

docker build -t kryshor/todo-frontend:1.2.0 ./frontend

docker build -t kryshor/todo-worker:1.2.0 ./worker

docker build -t kryshor/todo-nginx:1.2.0 ./nginx


# Subir imagenes a Docker Hub

docker push kryshor/todo-backend:1.2.0

docker push kryshor/todo-frontend:1.2.0

docker push kryshor/todo-worker:1.2.0

docker push kryshor/todo-nginx:1.2.0


# Listar imagenes

docker images | grep todo


# Eliminar imagen

docker rmi kryshor/todo-backend:1.2.0

 # Acceder a nodo master k3d node shell k3d-todo-cluster-server-0
```

Limpieza

Limpiar Docker Compose

```
# Detener todo y eliminar volumenes

docker compose down -v



# Eliminar imagenes

docker rmi $(docker images | grep "todo" | awk '{print $3}')
```

Informacion de Versionamiento

Version Actual: 1.2.0

Ultima actualizacion: 29 de Octubre de 2025

Historial de Cambios

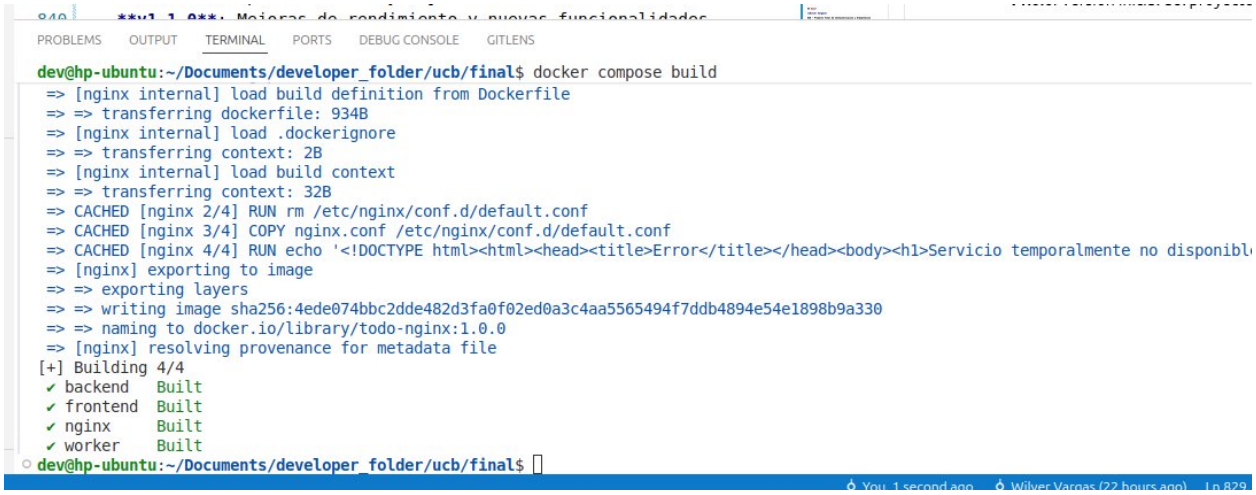
- **v1.2.0**: Optimizaciones y mejoras de estabilidad
- **v1.1.0**: Mejoras de rendimiento y nuevas funcionalidades
- **v1.0.0**: Version inicial del proyecto

Autor

Wilver Vargas

UCB - Proyecto Final de Containerizacion y Orquestacion

ANEXO CAPTURAS DE PANTALLA DE LA APLICACION



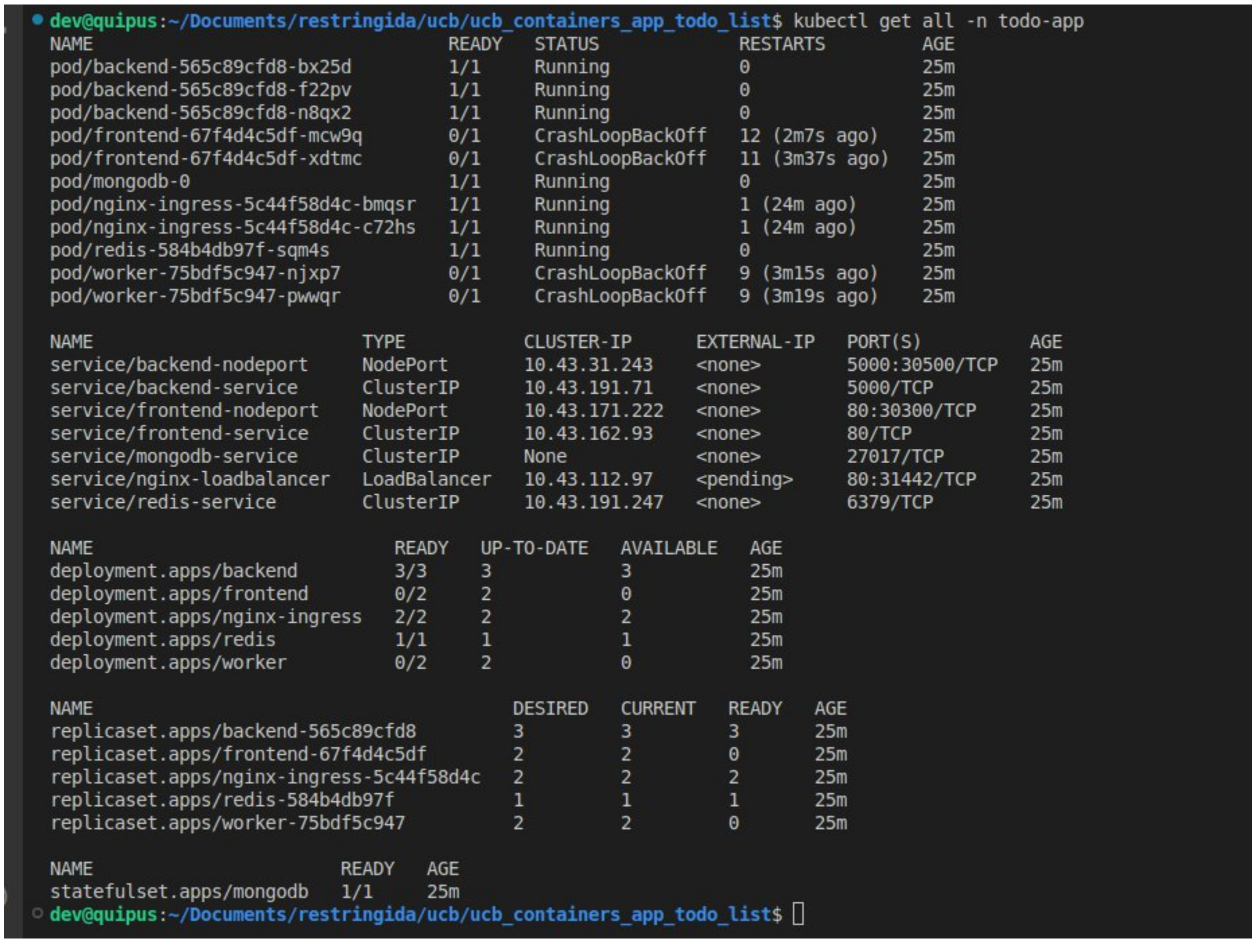
```
    **v1.1.0**: Mejoras de rendimiento y nuevas funcionalidades
```

PROBLEMS   OUTPUT   TERMINAL   PORTS   DEBUG CONSOLE   GITLENS

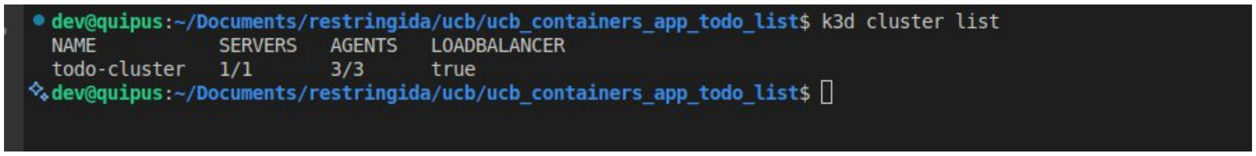
```
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ docker compose build
 => [nginx internal] load build definition from Dockerfile
 => => transferring dockerfile: 934B
 => [nginx internal] load .dockerignore
 => => transferring context: 2B
 => [nginx internal] load build context
 => => transferring context: 32B
 => CACHED [nginx 2/4] RUN rm /etc/nginx/conf.d/default.conf
 => CACHED [nginx 3/4] COPY nginx.conf /etc/nginx/conf.d/default.conf
 => CACHED [nginx 4/4] RUN echo '<!DOCTYPE html><html><head><title>Error</title></head><body><h1>Servicio temporalmente no disponibl
 => [nginx] exporting to image
 => => exporting layers
 => => writing image sha256:4ede074bbc2dde482d3fa0f02ed0a3c4aa5565494f7ddb4894e54e1898b9a330
 => => naming to docker.io/library/todo-nginx:1.0.0
 => [nginx] resolving provenance for metadata file
[+] Building 4/4
 ✔ backend    Built
 ✔ frontend   Built
 ✔ nginx      Built
 ✔ worker     Built
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ 
```

You, 1 second ago   Wilver Vargas (22 hours ago)   Ln 829

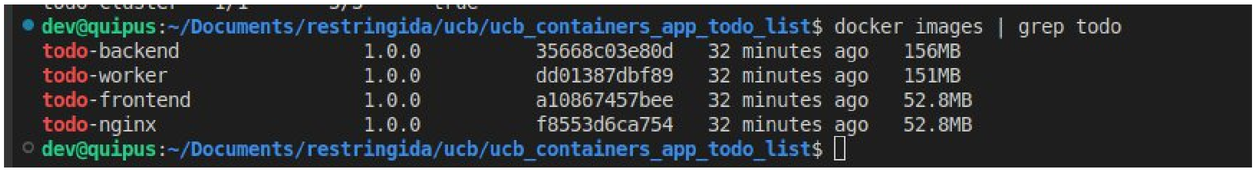
Construccion de la las imagenes



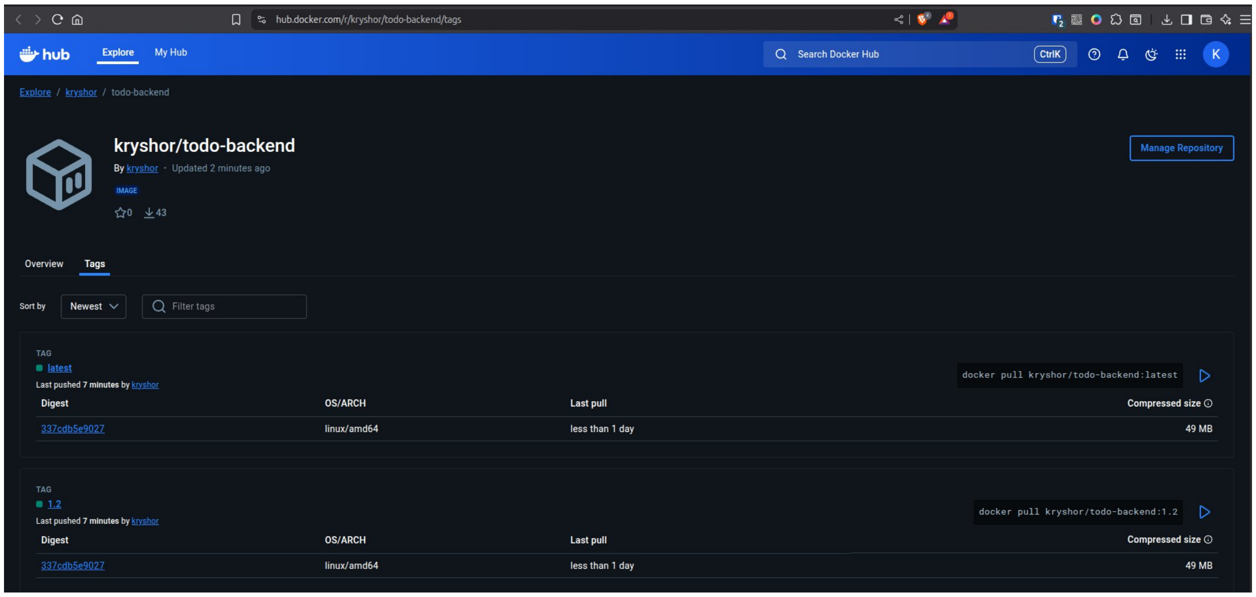
```
dev@quipus:~/Documents/restringida/ucb/ucb_containers_app_todo_list$ kubectl get all -n todo-app
NAME                                   READY   STATUS             RESTARTS         AGE
pod/backend-565c89cfd8-bx25d           1/1     Running            0                25m
pod/backend-565c89cfd8-f22pv           1/1     Running            0                25m
pod/backend-565c89cfd8-n8qx2           1/1     Running            0                25m
pod/frontend-67f4d4c5df-mcw9q          0/1     CrashLoopBackOff   12 (2m7s ago)    25m
pod/frontend-67f4d4c5df-xdtmc          0/1     CrashLoopBackOff   11 (3m37s ago)   25m
pod/mongodb-0                          1/1     Running            0                25m
pod/nginx-ingress-5c44f58d4c-bmqsr     1/1     Running            1 (24m ago)      25m
pod/nginx-ingress-5c44f58d4c-c72hs     1/1     Running            1 (24m ago)      25m
pod/redis-584b4db97f-sqm4s             1/1     Running            0                25m
pod/worker-75bdf5c947-njxp7            0/1     CrashLoopBackOff   9 (3m15s ago)    25m
pod/worker-75bdf5c947-pwwqr            0/1     CrashLoopBackOff   9 (3m19s ago)    25m

NAME                          TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)           AGE
service/backend-nodeport      NodePort       10.43.31.243    <none>        5000:30500/TCP    25m
service/backend-service       ClusterIP      10.43.191.71    <none>        5000/TCP          25m
service/frontend-nodeport     NodePort       10.43.171.222   <none>        80:30300/TCP      25m
service/frontend-service      ClusterIP      10.43.162.93    <none>        80/TCP            25m
service/mongodb-service       ClusterIP      None            <none>        27017/TCP         25m
service/nginx-loadbalancer    LoadBalancer   10.43.112.97    <pending>     80:31442/TCP      25m
service/redis-service         ClusterIP      10.43.191.247   <none>        6379/TCP          25m

NAME                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/backend         3/3     3            3           25m
deployment.apps/frontend        0/2     2            0           25m
deployment.apps/nginx-ingress   2/2     2            2           25m
deployment.apps/redis           1/1     1            1           25m
deployment.apps/worker          0/2     2            0           25m

NAME                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/backend-565c89cfd8         3         3         3       25m
replicaset.apps/frontend-67f4d4c5df        2         2         0       25m
replicaset.apps/nginx-ingress-5c44f58d4c   2         2         2       25m
replicaset.apps/redis-584b4db97f           1         1         1       25m
replicaset.apps/worker-75bdf5c947          2         2         0       25m

NAME                       READY   AGE
statefulset.apps/mongodb   1/1     25m
dev@quipus:~/Documents/restringida/ucb/ucb_containers_app_todo_list$ 
```



```
dev@quipus:~/Documents/restringida/ucb/ucb_containers_app_todo_list$ k3d cluster list
NAME           SERVERS   AGENTS   LOADBALANCER
todo-cluster   1/1       3/3      true
dev@quipus:~/Documents/restringida/ucb/ucb_containers_app_todo_list$ 
```



```
dev@quipus:~/Documents/restringida/ucb/ucb_containers_app_todo_list$ docker images | grep todo
todo-backend                1.0.0          35668c03e80d   32 minutes ago   156MB
todo-worker                 1.0.0          dd01387dbf89   32 minutes ago   151MB
todo-frontend               1.0.0          a10867457bee   32 minutes ago   52.8MB
todo-nginx                  1.0.0          f8553d6ca754   32 minutes ago   52.8MB
dev@quipus:~/Documents/restringida/ucb/ucb_containers_app_todo_list$ 
```



kryshor/todo-backend

By kryshor · Updated 2 minutes ago

0 43

TAG
latest
Last pushed 7 minutes by kryshor

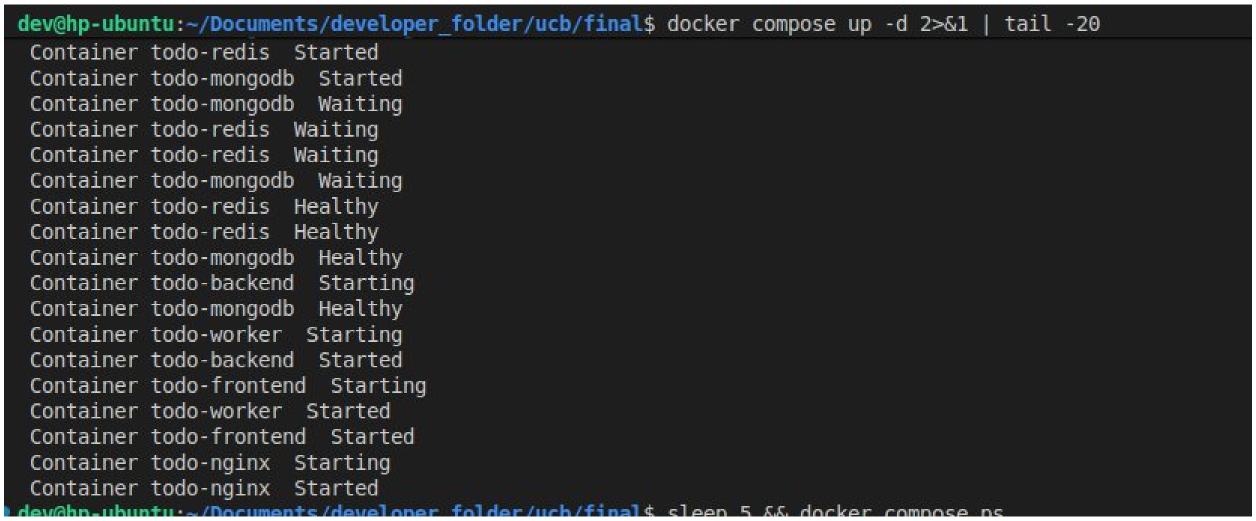
`docker pull kryshor/todo-backend:latest`

Digest	OS/ARCH	Last pull	Compressed size
337cdb5e9027	linux/amd64	less than 1 day	49 MB

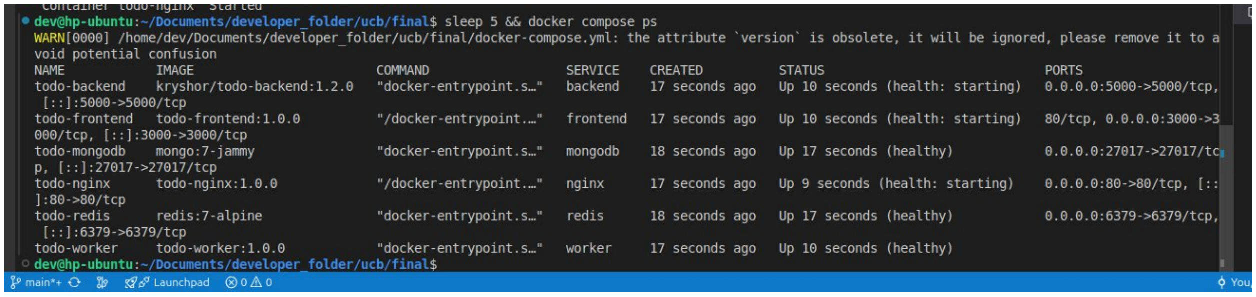
TAG
1.2
Last pushed 7 minutes by kryshor

`docker pull kryshor/todo-backend:1.2`

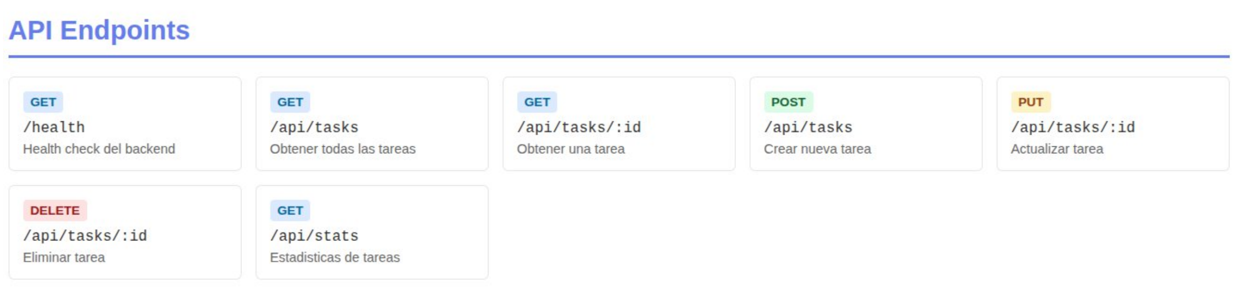
Digest	OS/ARCH	Last pull	Compressed size
337cdb5e9027	linux/amd64	less than 1 day	49 MB



```
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ docker compose up -d 2>&1 | tail -20
 Container todo-redis  Started
 Container todo-mongodb  Started
 Container todo-mongodb  Waiting
 Container todo-redis  Waiting
 Container todo-redis  Waiting
 Container todo-mongodb  Waiting
 Container todo-redis  Healthy
 Container todo-redis  Healthy
 Container todo-mongodb  Healthy
 Container todo-backend  Starting
 Container todo-mongodb  Healthy
 Container todo-worker  Starting
 Container todo-backend  Started
 Container todo-frontend  Starting
 Container todo-worker  Started
 Container todo-frontend  Started
 Container todo-nginx  Starting
 Container todo-nginx  Started
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ sleep 5 && docker compose ps
```



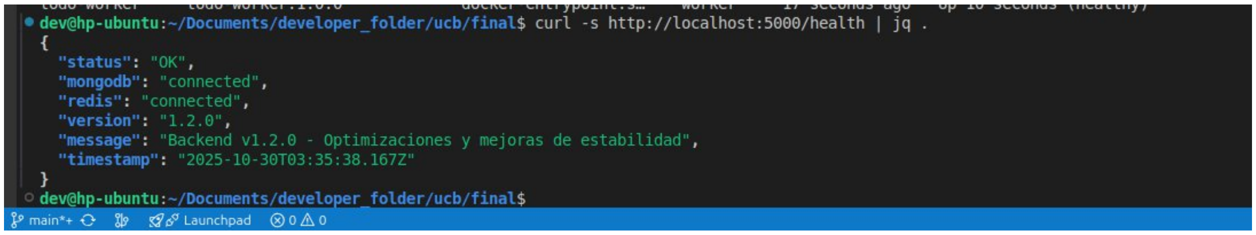
```
 Container todo-nginx  Started
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ sleep 5 && docker compose ps
WARN[0000] /home/dev/Documents/developer_folder/ucb/final/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to a
void potential confusion
NAME            IMAGE                         COMMAND                   SERVICE    CREATED          STATUS                             PORTS
todo-backend    kryshor/todo-backend:1.2.0    "docker-entrypoint.s…"    backend    17 seconds ago   Up 10 seconds (health: starting)   0.0.0.0:5000->5000/tcp,
 [::]:5000->5000/tcp
todo-frontend   todo-frontend:1.0.0           "/docker-entrypoint.…"    frontend   17 seconds ago   Up 10 seconds (health: starting)   80/tcp, 0.0.0.0:3000->3
000/tcp, [::]:3000->3000/tcp
todo-mongodb    mongo:7-jammy                 "docker-entrypoint.s…"    mongodb    18 seconds ago   Up 17 seconds (healthy)            0.0.0.0:27017->27017/tc
p, [::]:27017->27017/tcp
todo-nginx      todo-nginx:1.0.0              "/docker-entrypoint.…"    nginx      17 seconds ago   Up 9 seconds (health: starting)    0.0.0.0:80->80/tcp, [::
]:80->80/tcp
todo-redis      redis:7-alpine                "docker-entrypoint.s…"    redis      18 seconds ago   Up 17 seconds (healthy)            0.0.0.0:6379->6379/tcp,
 [::]:6379->6379/tcp
todo-worker     todo-worker:1.0.0             "docker-entrypoint.s…"    worker     17 seconds ago   Up 10 seconds (healthy)
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$
```



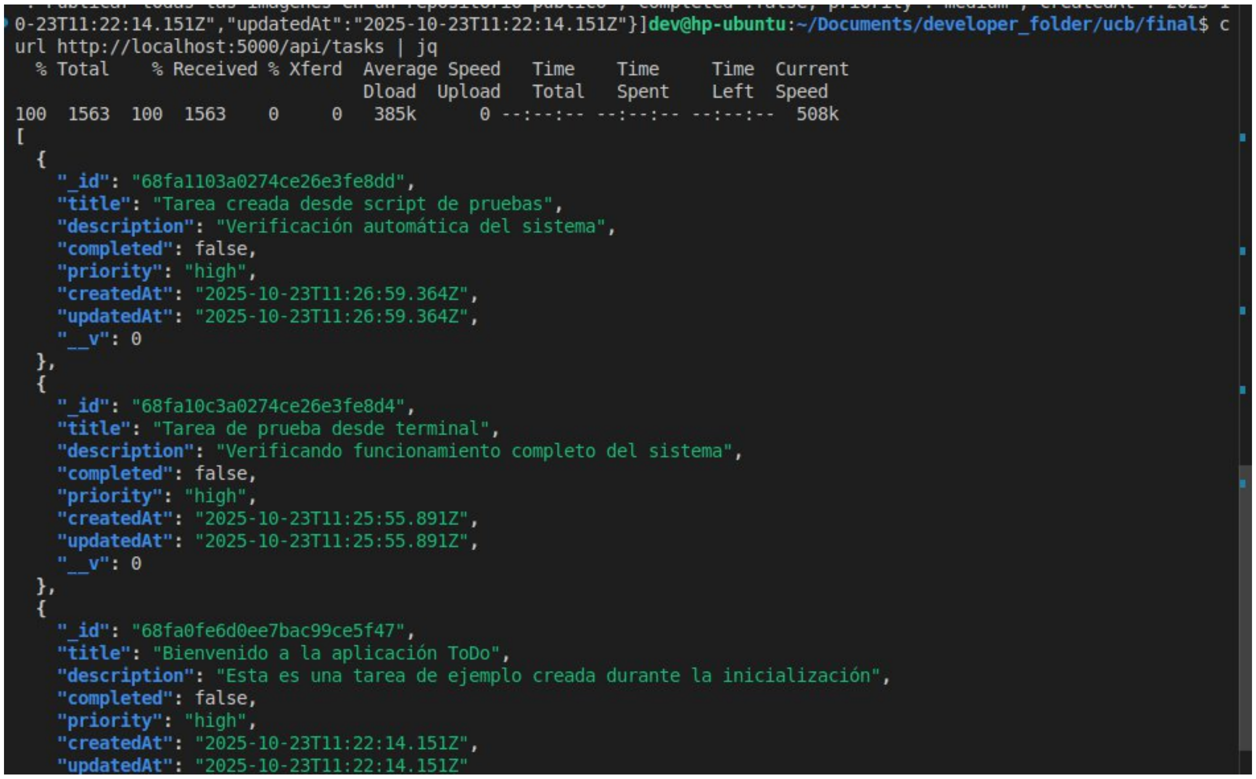
API Endpoints

GET	/health	Health check del backend
GET	/api/tasks	Obtener todas las tareas
GET	/api/tasks/:id	Obtener una tarea
POST	/api/tasks	Crear nueva tarea
PUT	/api/tasks/:id	Actualizar tarea
DELETE	/api/tasks/:id	Eliminar tarea
GET	/api/stats	Estadisticas de tareas

Programa funcional haciendo peticiones curl

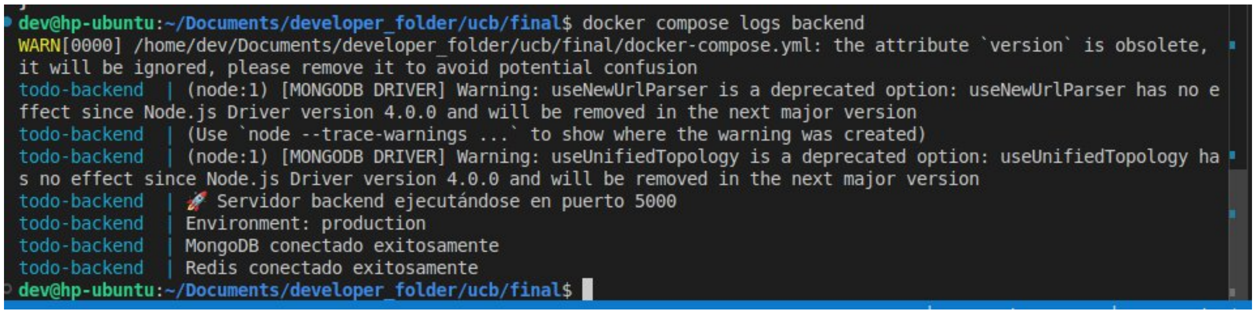


```
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ curl -s http://localhost:5000/health | jq .
{
  "status": "OK",
  "mongodb": "connected",
  "redis": "connected",
  "version": "1.2.0",
  "message": "Backend v1.2.0 - Optimizaciones y mejoras de estabilidad",
  "timestamp": "2025-10-30T03:35:38.167Z"
}
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$
```



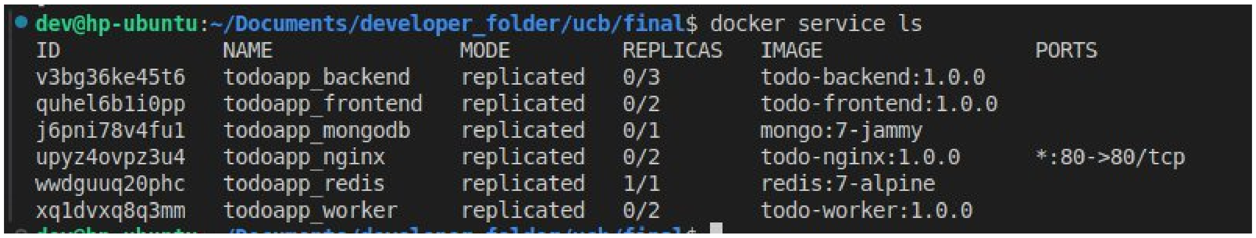
```
0-23T11:22:14.151Z","updatedAt":"2025-10-23T11:22:14.151Z"}]dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ c
url http://localhost:5000/api/tasks | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1563  100  1563    0     0   385k      0 --:--:-- --:--:-- --:--:--  508k
[
  {
    "_id": "68fa1103a0274ce26e3fe8dd",
    "title": "Tarea creada desde script de pruebas",
    "description": "Verificación automática del sistema",
    "completed": false,
    "priority": "high",
    "createdAt": "2025-10-23T11:26:59.364Z",
    "updatedAt": "2025-10-23T11:26:59.364Z",
    "__v": 0
  },
  {
    "_id": "68fa10c3a0274ce26e3fe8d4",
    "title": "Tarea de prueba desde terminal",
    "description": "Verificando funcionamiento completo del sistema",
    "completed": false,
    "priority": "high",
    "createdAt": "2025-10-23T11:25:55.891Z",
    "updatedAt": "2025-10-23T11:25:55.891Z",
    "__v": 0
  },
  {
    "_id": "68fa0fe6d0ee7bac99ce5f47",
    "title": "Bienvenido a la aplicación ToDo",
    "description": "Esta es una tarea de ejemplo creada durante la inicialización",
    "completed": false,
    "priority": "high",
    "createdAt": "2025-10-23T11:22:14.151Z",
    "updatedAt": "2025-10-23T11:22:14.151Z"
```

Ver logs de backend



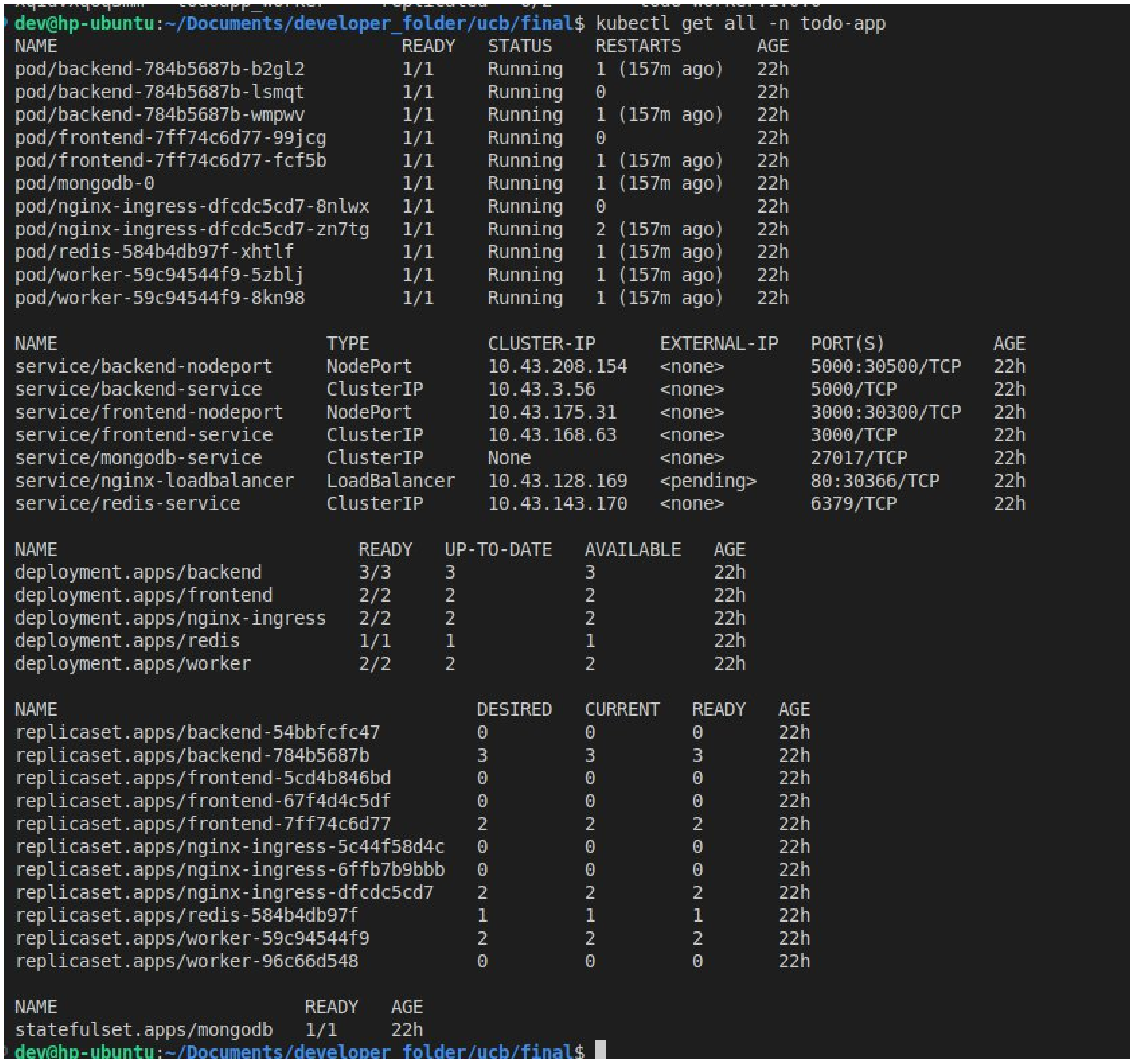
```
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ docker compose logs backend
WARN[0000] /home/dev/Documents/developer_folder/ucb/final/docker-compose.yml: the attribute `version` is obsolete,
it will be ignored, please remove it to avoid potential confusion
todo-backend  | (node:1) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no e
ffect since Node.js Driver version 4.0.0 and will be removed in the next major version
todo-backend  | (Use `node --trace-warnings ...` to show where the warning was created)
todo-backend  | (node:1) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology ha
s no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
todo-backend  | 🚀 Servidor backend ejecutándose en puerto 5000
todo-backend  | Environment: production
todo-backend  | MongoDB conectado exitosamente
todo-backend  | Redis conectado exitosamente
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$
```

LOGS DE DOCKER SWARM



```
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ docker service ls
ID             NAME                MODE         REPLICAS   IMAGE                  PORTS
v3bg36ke45t6   todoapp_backend     replicated   0/3        todo-backend:1.0.0
quhel6b1i0pp   todoapp_frontend    replicated   0/2        todo-frontend:1.0.0
j6pni78v4fu1   todoapp_mongodb     replicated   0/1        mongo:7-jammy
upyz4ovpz3u4   todoapp_nginx       replicated   0/2        todo-nginx:1.0.0       *:80->80/tcp
wwdguuq20phc   todoapp_redis       replicated   1/1        redis:7-alpine
xq1dvxq8q3mm   todoapp_worker      replicated   0/2        todo-worker:1.0.0
```

LOGS DE DOCKER KUBERNETES



```
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ kubectl get all -n todo-app
NAME                                 READY   STATUS    RESTARTS       AGE
pod/backend-784b5687b-b2gl2          1/1     Running   1 (157m ago)   22h
pod/backend-784b5687b-lsmqt          1/1     Running   0              22h
pod/backend-784b5687b-wmpwv          1/1     Running   1 (157m ago)   22h
pod/frontend-7ff74c6d77-99jcg        1/1     Running   0              22h
pod/frontend-7ff74c6d77-fcf5b        1/1     Running   1 (157m ago)   22h
pod/mongodb-0                        1/1     Running   1 (157m ago)   22h
pod/nginx-ingress-dfcdc5cd7-8nlwx    1/1     Running   0              22h
pod/nginx-ingress-dfcdc5cd7-zn7tg    1/1     Running   2 (157m ago)   22h
pod/redis-584b4db97f-xhtlf           1/1     Running   1 (157m ago)   22h
pod/worker-59c94544f9-5zblj          1/1     Running   1 (157m ago)   22h
pod/worker-59c94544f9-8kn98          1/1     Running   1 (157m ago)   22h

NAME                            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/backend-nodeport        NodePort       10.43.208.154   <none>        5000:30500/TCP   22h
service/backend-service         ClusterIP      10.43.3.56      <none>        5000/TCP         22h
service/frontend-nodeport       NodePort       10.43.175.31    <none>        3000:30300/TCP   22h
service/frontend-service        ClusterIP      10.43.168.63    <none>        3000/TCP         22h
service/mongodb-service         ClusterIP      None            <none>        27017/TCP        22h
service/nginx-loadbalancer      LoadBalancer   10.43.128.169   <pending>     80:30366/TCP     22h
service/redis-service           ClusterIP      10.43.143.170   <none>        6379/TCP         22h

NAME                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/backend          3/3     3            3           22h
deployment.apps/frontend         2/2     2            2           22h
deployment.apps/nginx-ingress    2/2     2            2           22h
deployment.apps/redis            1/1     1            1           22h
deployment.apps/worker           2/2     2            2           22h

NAME                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/backend-54bbfcfc47         0         0         0       22h
replicaset.apps/backend-784b5687b          3         3         3       22h
replicaset.apps/frontend-5cd4b846bd        0         0         0       22h
replicaset.apps/frontend-67f4d4c5df        0         0         0       22h
replicaset.apps/frontend-7ff74c6d77        2         2         2       22h
replicaset.apps/nginx-ingress-5c44f58d4c   0         0         0       22h
replicaset.apps/nginx-ingress-6ffb7b9bbb   0         0         0       22h
replicaset.apps/nginx-ingress-dfcdc5cd7    2         2         2       22h
replicaset.apps/redis-584b4db97f           1         1         1       22h
replicaset.apps/worker-59c94544f9          2         2         2       22h
replicaset.apps/worker-96c66d548           0         0         0       22h

NAME                       READY   AGE
statefulset.apps/mongodb   1/1     22h
dev@hp-ubuntu:~/Documents/developer_folder/ucb/final$ 
```