

# Kubernetes Taller 2

## 1. Dashboard

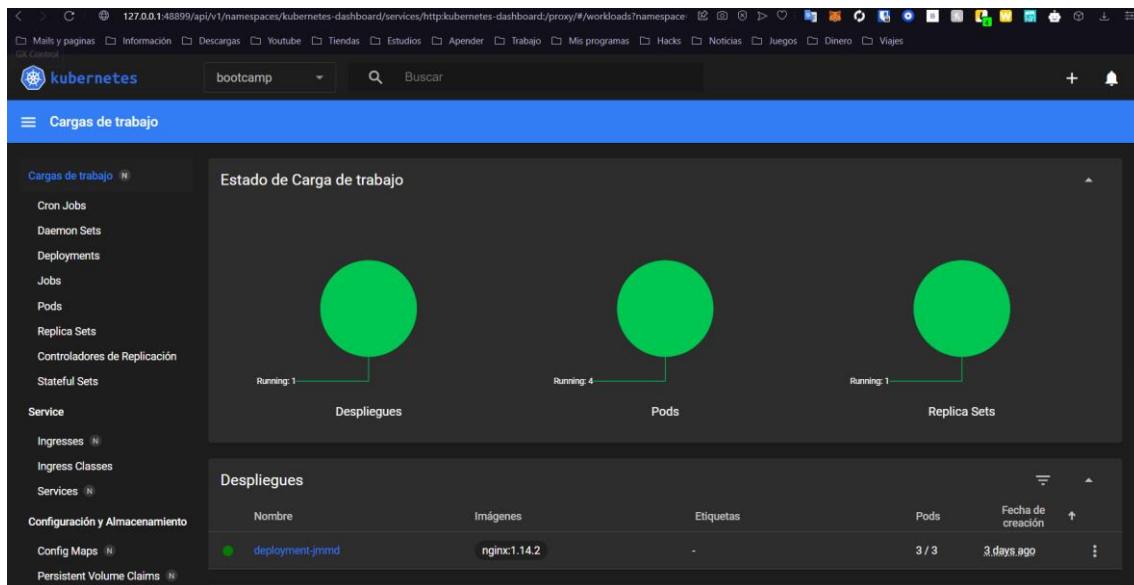
Para abrir el Dashboard con minikube, primero lo iniciamos y luego lo abrimos con el comando minikube dashboard.

```
C:\Users\josem>minikube start
* minikube v1.28.0 en Microsoft Windows 11 Pro For Workstations 10.0.22621 Build 22621
* Using the hyperv driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Restarting existing hyperv VM for "minikube" ...
* Preparando Kubernetes v1.25.3 en Docker 20.10.20...
* Verifying Kubernetes components...
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube addons enable metrics-server

* Complementos habilitados: storage-provisioner, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

C:\Users\josem>minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:48899/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in yo
ur default browser...
```



## 2. Secret

Primero creamos el archivo yaml. La contraseña la he encriptado en base64.

```
C: > Users > josem > Desktop > Desarrollo > NTT DATA > Bootcamp > App > nttdatacenter-kubernetes-t2-jmmd > ! secret-jmmd.yaml
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: jmmd-app
5  data:
6    POSTGRES_PASSWORD: UzBONFJRVUJF
7  stringData:
8    POSTGRES_USER: sonar_user
```

Una vez creado, lo aplicamos. En mi caso lo he aplicado al namespace que ya teníamos de bootcamp. Podemos comprobar que se ha aplicado correctamente.

```
C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatacenter-kubernetes-t2-jmmd>kubectl apply -f secret-jmmd.yaml -n bootcamp
secret/secret-jmmd created

C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatacenter-kubernetes-t2-jmmd>kubectl get secrets -n bootcamp
```

NAME	TYPE	DATA	AGE
secret-jmmd	Opaque	2	3s

### 3. ConfigMap

De nuevo, creamos el ConfigMap como archivo yaml.

```
C: > Users > josem > Desktop > Desarrollo > NTT DATA > Bootcamp > App > nttdatacenter-kubernetes-t2-jmmd > ! configmap-jmmd.yaml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: jmmd-app
5  data:
6    POSTGRES_DB: sonar_db
```

Una vez creado, lo aplicamos a nuestro namespace.

```
C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatacenter-kubernetes-t2-jmmd>kubectl apply -f configmap-jmmd.yaml
configmap/configmap-jmmd created
```

### 4. Persistent Volumen y Persistent Volume Claim

Creamos los archivos yaml.

- Persistent Volume Claim

```
C: > Users > josem > Desktop > Desarrollo > NTT DATA > Bootcamp > App > nttdatacenter-kubernetes-t2-jmmd > ! persistentclaim-jmmd.yaml
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: pvc-jmmd
5    labels:
6      app: jmmd-app
7  spec:
8    accessModes:
9      - ReadWriteOnce
10   resources:
11     requests:
12       storage: 3Gi
```

```
C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatacenter-kubernetes-t2-jmmd>kubectl apply -f persistentclaim-jmmd.yaml -n bootcamp
persistentvolumeclaim/pvc-jmmd unchanged
```

## - Persistent Volume

```
C: > Users > josem > Desktop > Desarrollo > NTT DATA > Bootcamp > App > nttdatacenter-kubernetes-t2-jmmd > ! persistent-jmmd.yaml
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: pv-jmmd
5    labels:
6      type: local
7  spec:
8    storageClassName: manual
9    hostPath:
10     path: "/mnt/data"
11   accessModes:
12     - ReadWriteOnce
13   capacity:
14     storage: 5Gi
```

```
C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatacenter-kubernetes-t2-jmmd>kubectl apply -f persistent-jmmd.yaml -n bootcamp
persistentvolume/pv-jmmd created
```

## 5. Deployment

Creamos el archivo yaml y le indicamos los que hemos creado anteriormente.

```

C: > Users > josem > Desktop > Desarrollo > NTT DATA > Bootcamp > App > nttdatcenter-kubernetes-t2-jmmd > ! deployment-jmmd.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: jmmd-app
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: jmmd-app
10   template:
11     metadata:
12       labels:
13         app: jmmd-app
14     spec:
15       containers:
16         - name: postgres
17           image: postgres:14.6
18           ports:
19             - containerPort: 5432
20           envFrom:
21             - secretRef:
22               name: jmmd-app
23             - configMapRef:
24               name: jmmd-app
25           volumeMounts:
26             - name: postgres-database-storage
27               mountPath: /var/lib/pgsql/data
28           volumes:
29             - name: postgres-database-storage
30               persistentVolumeClaim:
31                 claimName: pvc-jmmd

```

```

C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-kubernetes-t2-jmmd>kubectl apply -f deployment-jmmd.yaml -n bootcamp
deployment.apps/jmmd-app created

```

```

C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-kubernetes-t2-jmmd>kubectl get deployments -n bootcamp

```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment-jmmd	3/3	3	3	3d21h
jmmd-app	1/1	1	1	7m12s

Ahora comprobamos si se puede entrar en Postgres.

```

C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-kubernetes-t2-jmmd>kubectl exec -it jmmd-app-5d5fddc98-4h8g9 -n bootcamp -- psql -h localhost -U sonar_user sonar_db
psql (14.6 (Debian 14.6-1.pgdg110+1))
Type "help" for help.

sonar_db=#

```

## 6. Rolling update

Primero creamos el deployment.yaml con la nueva versión.

```

C: > Users > josem > Desktop > Desarrollo > NTT DATA > Bootcamp > App > nttdatcenter-kubernetes-t2-jmmd > ! newversion_deployment-jmmd.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: jmmd-app
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: jmmd-app
10   template:
11     metadata:
12       labels:
13         app: jmmd-app
14     spec:
15       containers:
16       - name: postgres
17         image: postgres:15.1
18         ports:
19         - containerPort: 5432
20         envFrom:
21         - secretRef:
22             name: jmmd-app
23         - configMapRef:
24             name: jmmd-app
25         volumeMounts:
26         - name: postgres-database-storage
27           mountPath: /var/lib/pgsql/data
28       volumes:
29       - name: postgres-database-storage
30         persistentVolumeClaim:
31           claimName: pvc-jmmd

```

Ahora exportamos los datos de nuestra antigua versión al archivo dumpfile.

```

C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-kubernetes-t2-jmmd>kubectl exec -it jmmd-app-7fbd74679b-mvnh -- /usr/bin/pg_dumpall -U sonar_user > dumpfile

```

```

dumpfile: Bloc de notas
Archivo  Editar  Ver

--
-- PostgreSQL database cluster dump
--

SET default_transaction_read_only = off;

SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;

--
-- Roles
--

CREATE ROLE sonar_user;

```

Instalamos el nuevo deployment

```

C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-kubernetes-t2-jmmd>kubectl apply -f newversion_deployment-jmmd.yaml -n bootcamp
deployment.apps/jmmd-app created

C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatcenter-kubernetes-t2-jmmd>kubectl get pods -n bootcamp
NAME                                READY   STATUS    RESTARTS   AGE
deployment-jmmd-74ff6b46f-5cpfb     1/1     Running   1 (3d22h ago)    3d23h
deployment-jmmd-74ff6b46f-dwfrf     1/1     Running   1 (3d22h ago)    3d23h
deployment-jmmd-74ff6b46f-xglnf     1/1     Running   1 (3d22h ago)    3d23h
jmmd-app-7fbd74679b-nl4jt           1/1     Running   0           11s
nginx                                1/1     Running   1 (3d22h ago)    4d3h

```

Importamos el archivo anteriormente creado

```
C:\Users\josem\Desktop\Desarrollo\NTT DATA\Bootcamp\App\nttdatacenter-kubernetes-t2-jmmd>kubectl exec -it jmmd-app-5d5fdcdc98-vjbmz -n bootcamp -- psql < dumpfile
Unable to use a TTY - input is not a terminal or the right kind of file
psql: error: connection to server on socket "/var/run/postgresql/.s.PGSQL.5432" failed: FATAL:  role "root" does not exist
command terminated with exit code 2
```

Nota: a la hora de importar el nuevo sql ha surgido un error el cual creo que procede del tema de usuarios/contraseñas que vimos esta mañana, porque por más que miro no encuentro como corregirlo.