

Uso de Kubernetes

Creando un cluster Kubernetes

Instalar kubectl y Minikube según la bitácora en el sitio de la cátedra.

Agregando servicios al cluster

Para crear un servicio utilizar los siguientes comandos:

```
kubectl run hello-minikube --image=k8s.gcr.io/echoserver:1.4 \  
--port=8080
```

```
kubectl expose deployment hello-minikube --type=NodePort
```

Qué hace cada uno de estos comandos?

Podemos observar la orquestación del servicio ejecutando de manera repetida el comando:

```
kubectl get pod
```

Una vez que el pod esté en estado 'Running' podemos ver la URL de acceso al servicio con el comando:

```
minikube service hello-minikube --url
```

Accedemos al servicio usando cURL:

```
curl $(minikube service hello-minikube --url)
```

También se puede emplear un navegador web.

Eliminando servicios

Para eliminar el servicio creado emplear los comandos:

```
kubectl delete service hello-minikube
```

```
kubectl delete deployment hello-minikube
```

Correr una aplicación distribuida en el cluster

Para esto vamos a emplear un proyecto de ejemplo disponible [aquí](#). En el mismo lugar podremos ver una descripción de los containers que se crearán, sus funciones y como se relacionan entre sí.

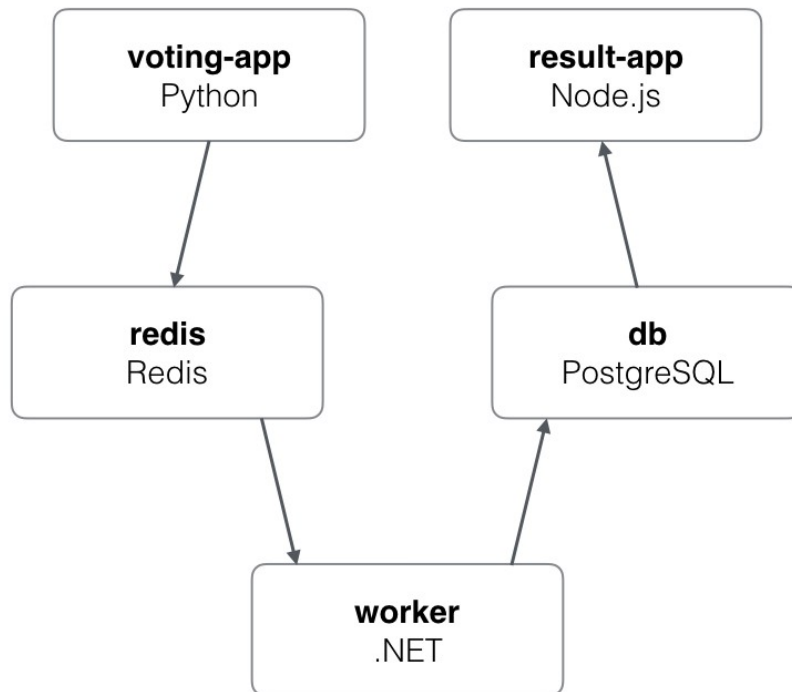


Ilustración 1: Arquitectura del Stack

Clonamos el repositorio con el stack usando git:

```
git clone https://github.com/docker-samples/example-voting-app.git
```

Ingresamos al directorio recién creado

```
cd example-voting-app/
```

Dentro del directorio encontraremos lo necesario para desplegar este stack utilizando Docker Compose o Google Kubernetes.

Antes de crear los elementos del stack crearemos un namespace llamado **vote** para agruparlos:

```
kubectl create namespace vote
```

La carpeta **k8s-specifications** contiene las definiciones en YAML de los servicios y deployments del stack.

Creamos los servicios y deployments usando kubectl a partir de los archivos YAML
kubectl create -f k8s-specifications/

Veremos la creación y el estado de los pods, servicios, deployments y réplicas mediante:

kubectl get all --namespace vote

Ejecutando este comando de manera reiterada veremos como se van creando e iniciando los distintos elementos y como los replicaset van llegando a su estado deseado.

Si queremos más información sobre cada elemento podemos usar:

kubectl get all --namespace vote -o wide

Para ver las URLs de los servicios usaremos:

minikube service vote --url --namespace vote

y

minikube service result --url --namespace vote

Si visitamos esas páginas con diferentes navegadores podremos efectuar votos (el servicio **vote**) y ver los resultados (servicio **result**)

Si deseáramos contar con más réplicas de un deployment lo podemos modificar con:

kubectl scale --replicas=3 deployment vote --namespace vote

En este caso hacemos que el deployment vote tenga tres réplicas.

Podemos verificar el cambio listando los pods y replicaset:

kubectl get pods,replicaset --namespace vote