

Tutoriales

- 1: [Hello Minikube](#)
- 2: [Introducción](#)
 - 2.1: [Crear un clúster](#)
 - 2.1.1: [Tutorial interactivo - Crear un clúster](#)
 - 2.2: [Desplegar una aplicación](#)
 - 2.2.1: [Tutorial interactivo - Implementando una aplicación](#)
 - 2.3: [Consultar el estado de una aplicación](#)
 - 2.4: [Exponer una aplicación](#)
 - 2.5: [Escalar una aplicación](#)
 - 2.6: [Actualizar una aplicación](#)
- 3: [Formación en línea](#)
- 4: [Configuración](#)
- 5: [Aplicaciones Stateless](#)
- 6: [Aplicaciones Stateful](#)
- 7: [Clústeres](#)
- 8: [Servicios](#)

ón de la documentación de Kubernetes contiene tutoriales.

l muestra cómo lograr una meta que es más grande que una sola [tarea \(/docs/tasks/\)](#).
te un tutorial tiene varias secciones y cada una de ellas contiene un procedimiento. Antes de
ada tutorial, recomendamos añadir un marcador a [Glosario de términos](#)
[erence/glossary/](#)) para poder consultarlo fácilmente.

iales

[Kubernetes Basics \(/docs/tutorials/kubernetes-basics/\)](#) se trata de un tutorial interactivo en
indidad para entender Kubernetes y probar algunas funciones básicas.

[Scalable Microservices with Kubernetes \(Udacity\)](#) (<https://www.udacity.com/course/scalable-microservices-with-kubernetes--ud615>).

[Introduction to Kubernetes \(edX\)](#) (<https://www.edx.org/course/introduction-kubernetes-foundationx-lfs158x#>).

[Hello Minikube \(/es/docs/tutorials/hello-minikube/\)](#).

guración

[Tutorial: Configurando un Microservicio en Java \(/docs/tutorials/configuration/configure-java-service/\)](#).

[Tutorial: Configuring Redis Using a ConfigMap \(/docs/tutorials/configuration/configure-redis-using-configmap/\)](#).

aciones Stateless

[Tutorial: Exposing an External IP Address to Access an Application in a Cluster \(/docs/tutorials/stateless-application/expose-external-ip-address/\)](#).

[Tutorial: Deploying PHP Guestbook application with Redis \(/docs/tutorials/stateless-application/guestbook/\)](#).

aciones Stateful

[Tutorial: StatefulSet Basics \(/docs/tutorials/stateful-application/basic-stateful-set/\)](#).

[Tutorial: Deploying WordPress and MySQL with Persistent Volumes \(/docs/tutorials/stateful-application/mysql-wordpress-persistent-volume/\)](#).

[Example: Deploying Cassandra with Stateful Sets \(/docs/tutorials/stateful-application/cassandra/\)](#)
[Deploying ZooKeeper, A CP Distributed System \(/docs/tutorials/stateful-application/zookeeper/\)](#)

ers

[AppArmor \(/docs/tutorials/clusters/apparmor/\)](#)
[Seccomp \(/docs/tutorials/clusters/seccomp/\)](#)

ios

[Using Source IP \(/docs/tutorials/services/source-ip/\)](#)

antes pasos

escribir un tutorial, revisa [utilizando templates \(/docs/home/contribute/page-templates/\)](#) para formación sobre el tipo de página y la plantilla de los tutotriales.

1 - Hello Minikube

Este tutorial muestra como ejecutar una aplicación Node.js Hola Mundo en Kubernetes utilizando [Minikube \(/docs/setup/learning-environment/minikube\)](https://docs.setup/learning-environment/minikube) y Katacoda. Katacoda provee un ambiente de Kubernetes desde el navegador.

Nota: También se puede seguir este tutorial si se ha instalado [Minikube localmente \(/docs/tasks/tools/install-minikube/\)](https://docs/tasks/tools/install-minikube/).

Objetivos

- Desplegar una aplicación Hola Mundo en Minikube.
- Ejecutar la aplicación.
- Ver los logs de la aplicación.

Antes de empezar

Este tutorial provee una imagen de contenedor construida desde los siguientes archivos:

[minikube/server.js](https://raw.githubusercontent.com/kubernetes/website/main/content/es/examples/minikube/server.js)
(<https://raw.githubusercontent.com/kubernetes/website/main/content/es/examples/minikube/server.js>)

```
var http = require('http');

var handleRequest = function(request, response) {
  console.log('Received request for URL: ' + request.url);
  response.writeHead(200);
  response.end('Hello World!');
};

var www = http.createServer(handleRequest);
www.listen(8080);
```

[minikube/Dockerfile](https://raw.githubusercontent.com/kubernetes/website/main/content/es/examples/minikube/Dockerfile)
(<https://raw.githubusercontent.com/kubernetes/website/main/content/es/examples/minikube/Dockerfile>)

```
FROM node:6.14.2
EXPOSE 8080
COPY server.js .
CMD [ "node", "server.js" ]
```

Para más información sobre el comando `docker build` , lea la [documentación de Docker \(https://docs.docker.com/engine/reference/commandline/build/\)](https://docs.docker.com/engine/reference/commandline/build/).

Crear un clúster Minikube

1. Haz clic en **Launch Terminal**

Launch Terminal

Nota: Si se tiene instalado Minikube local, ejecutar `minikube start` .

2. Abrir el tablero de Kubernetes dashboard en un navegador:

```
minikube dashboard
```

- 3. Solo en el ambiente de Katacoda: En la parte superior de la terminal, haz clic en el símbolo + y luego clic en **Select port to view on Host 1**.
- 4. Solo en el ambiente de Katacoda: Escribir `30000` , y hacer clic en **Display Port**.

Crear un Deployment

Un *Pod* (</docs/concepts/workloads/pods/pod/>) en Kubernetes es un grupo de uno o más contenedores, asociados con propósitos de administración y redes. El Pod en este tutorial tiene solo un contenedor. Un *Deployment* (</docs/concepts/workloads/controllers/deployment/>) en Kubernetes verifica la salud del Pod y reinicia su contenedor si este es eliminado. Los Deployments son la manera recomendada de manejar la creación y escalación.

- 1. Ejecutar el comando `kubectl create` para crear un Deployment que maneje un Pod. El Pod ejecuta un contenedor basado en la imagen proveida por Docker.

```
kubectl create deployment hello-node --image=k8s.gcr.io/echoserver:1.4
```

- 2. Ver el Deployment:

```
kubectl get deployments
```

El resultado es similar a:

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
hello-node	1/1	1	1	1m

- 3. Ver el Pod:

```
kubectl get pods
```

El resultado es similar a:

NAME	READY	STATUS	RESTARTS	AGE
hello-node-5f76cf6ccf-br9b5	1/1	Running	0	1m

- 4. Ver los eventos del clúster:

```
kubectl get events
```

- 5. Ver la configuración `kubectl` :

```
kubectl config view
```

Nota: Para más información sobre el comando `kubectl` , ver [kubectl overview \(/docs/user-guide/kubectl-overview/\)](/docs/user-guide/kubectl-overview/).

Crear un Service

Por defecto, el Pod es accedido por su dirección IP interna dentro del clúster de Kubernetes, para hacer que el contenedor `hello-node` sea accesible desde afuera de la red virtual Kubernetes, se debe exponer el Pod como un [Service \(/docs/concepts/services-networking/service/\)](/docs/concepts/services-networking/service/) de Kubernetes.

- 1. Exponer el Pod a la red pública de internet utilizando el comando `kubect1 expose :`

```
kubect1 expose deployment hello-node --type=LoadBalancer --port=8080
```

El flag `--type=LoadBalancer` indica que se quiere exponer el Service fuera del clúster.

- 2. Ver el Service creado:

```
kubect1 get services
```

El resultado es similar a:

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
hello-node	LoadBalancer	10.108.144.78	<pending>	8080:30369/TCP	21s
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	23m

Para los proveedores Cloud que soportan balanceadores de carga, una dirección IP externa será provisionada para acceder al servicio, en Minikube, el tipo `LoadBalancer` permite que el servicio sea accesible a través del comando `minikube service .`

- 3. Ejecutar el siguiente comando:

```
minikube service hello-node
```

- 4. Solo en el ambiente de Katacoda: Hacer clic sobre el símbolo +, y luego en **Select port to view on Host 1.**
- 5. Solo en el ambiente de Katacoda: Anotar el puerto de 5 dígitos ubicado al lado del valor de `8080` en el resultado de servicios. Este número de puerto es generado aleatoriamente y puede ser diferente al indicado en el ejemplo. Escribir el número de puerto en el cuadro de texto y hacer clic en Display Port. Usando el ejemplo anterior, usted escribiría `30369` .

Esto abre una ventana de navegador que contiene la aplicación y muestra el mensaje "Hello World".

Habilitar Extensiones

Minikube tiene un conjunto de [Extensiones \(/docs/concepts/cluster-administration/addons/\)](/docs/concepts/cluster-administration/addons/) que pueden ser habilitados y desahabilitados en el ambiente local de Kubernetes.

- 1. Listar las extensiones soportadas actualmente:

```
minikube addons list
```

El resultado es similar a:

```
addon-manager: enabled
dashboard: enabled
default-storageclass: enabled
efk: disabled
freshpod: disabled
gvisor: disabled
helm-tiller: disabled
ingress: disabled
ingress-dns: disabled
logviewer: disabled
metrics-server: disabled
nvidia-driver-installer: disabled
nvidia-gpu-device-plugin: disabled
registry: disabled
registry-creds: disabled
storage-provisioner: enabled
storage-provisioner-gluster: disabled
```

2. Habilitar una extensión, por ejemplo, `metrics-server` :

```
minikube addons enable metrics-server
```

El resultado es similar a:

```
metrics-server was successfully enabled
```

3. Ver el Pod y Service creados:

```
kubectl get pod,svc -n kube-system
```

El resultado es similar a:

NAME	READY	STATUS	RESTARTS	AGE
pod/coredns-5644d7b6d9-mh9ll	1/1	Running	0	34m
pod/coredns-5644d7b6d9-pqd2t	1/1	Running	0	34m
pod/metrics-server-67fb648c5	1/1	Running	0	26s
pod/etcd-minikube	1/1	Running	0	34m
pod/influxdb-grafana-b29w8	2/2	Running	0	26s
pod/kube-addon-manager-minikube	1/1	Running	0	34m
pod/kube-apiserver-minikube	1/1	Running	0	34m
pod/kube-controller-manager-minikube	1/1	Running	0	34m
pod/kube-proxy-rnlps	1/1	Running	0	34m
pod/kube-scheduler-minikube	1/1	Running	0	34m
pod/storage-provisioner	1/1	Running	0	34m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
service/metrics-server	ClusterIP	10.96.241.45	<none>	80/TCP
service/kube-dns	ClusterIP	10.96.0.10	<none>	53/UDP,53/TCP
service/monitoring-grafana	NodePort	10.99.24.54	<none>	80:30002/TCP
service/monitoring-influxdb	ClusterIP	10.111.169.94	<none>	8083/TCP,8086/TCP

4. Deshabilitar `metrics-server` :

```
minikube addons disable metrics-server
```

El resultado es similar a:

```
metrics-server was successfully disabled
```

Limpieza

Ahora se puede eliminar los recursos creados en el clúster:

```
kubect1 delete service hello-node  
kubect1 delete deployment hello-node
```

Opcional, detener la máquina virtual de Minikube:

```
minikube stop
```

Opcional, eliminar la máquina virtual de Minikube:

```
minikube delete
```

Siguientes pasos

- Leer más sobre [Deployments \(/docs/concepts/workloads/controllers/deployment/\)](/docs/concepts/workloads/controllers/deployment/).
- Leer más sobre [Desplegando aplicaciones \(/docs/tasks/run-application/run-stateless-application-deployment/\)](/docs/tasks/run-application/run-stateless-application-deployment/).
- Leer más sobre [Services \(/docs/concepts/services-networking/service/\)](/docs/concepts/services-networking/service/).

2 - Introducción

2.1 - Crear un clúster

2.1.1 - Tutorial interactivo - Crear un clúster

Este escenario interactivo es desplegar un clúster de desarrollo local utilizando minikube

La pantalla es un entorno Linux preconfigurado que se puede usar como consola normal (puede escribir comandos). Al hacer clic en los botones seguidos del signo INTRO, se ejecutará ese comando en el

Congratulations on completing this module

Want to hear what you thought of the module?

[Click here](#) in Google to save what you have done.

[Click here](#)

Este escenario interactivo es desplegar un clúster de desarrollo local utilizando minikube

La pantalla es un entorno Linux preconfigurado que se puede usar como consola normal (puede escribir comandos). Al hacer clic en los botones seguidos del signo INTRO, se ejecutará ese comando en el

Congratulations on completing this module

Want to hear what you thought of the module?

[Click here](#) in Google to save what you have done.

[Click here](#)

Este escenario interactivo es desplegar un clúster de desarrollo local utilizando minikube

La pantalla es un entorno Linux preconfigurado que se puede usar como consola normal (puede escribir comandos). Al hacer clic en los botones seguidos del signo INTRO, se ejecutará ese comando en el entorno.



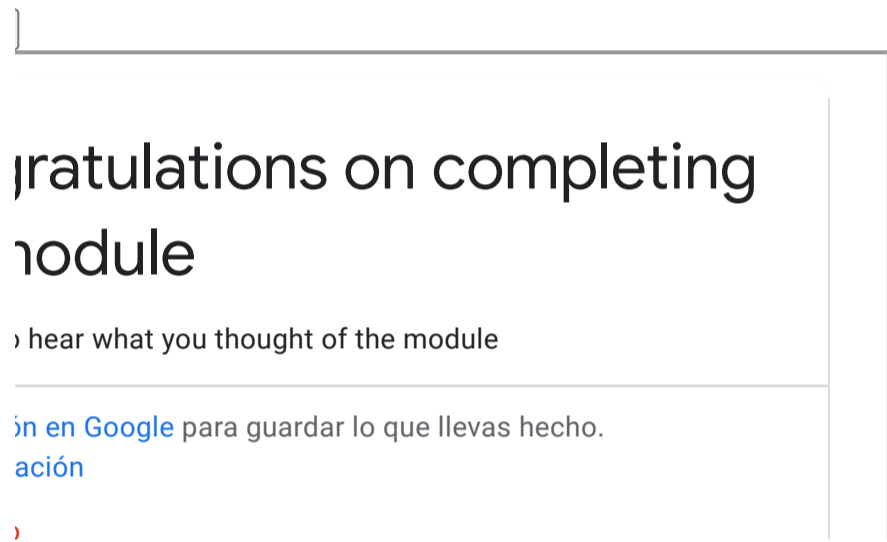
En el Módulo 2 ➤ [./docs/tutorials/kubernetes-basics/deploy-app/deploy-intro/](https://docs/tutorials/kubernetes-basics/deploy-app/deploy-intro/)

2.2 - Desplegar una aplicación

2.2.1 - Tutorial interactivo - Implementando una aplicación

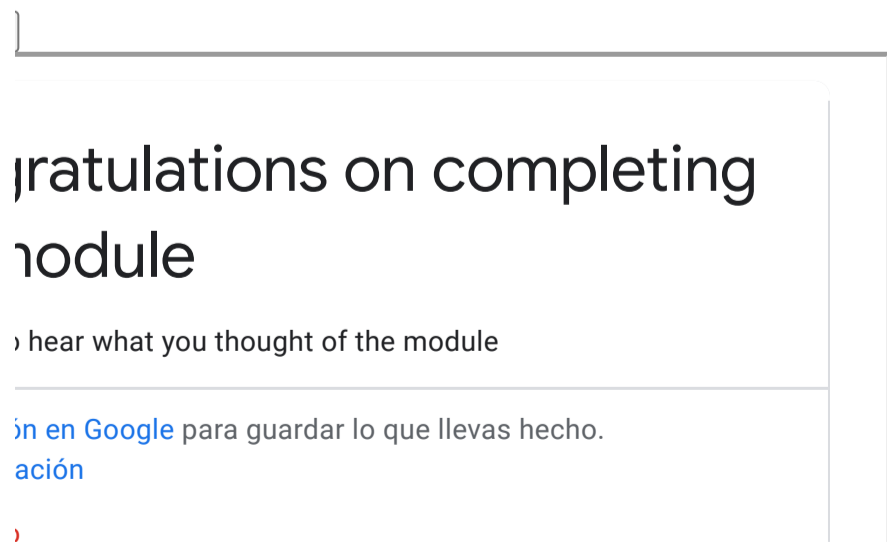
Este escenario es ayudarte a desplegar tu primera aplicación en Kubernetes usando kubectl. Aprenderás lo principal sobre el cli y como interactuar con tu aplicacion

un entorno Linux pre-configurado que puede ser usada como una
describir comandos). Haciendo click en los bloques de código
el comando se ejecutará en la terminal.



Este escenario es ayudarte a desplegar tu primera aplicación en Kubernetes usando kubectl. Aprenderás lo principal sobre el cli y como interactuar con tu aplicacion

un entorno Linux pre-configurado que puede ser usada como una
describir comandos). Haciendo click en los bloques de código
el comando se ejecutará en la terminal.



Este escenario es ayudarte a desplegar tu primera aplicación en Kubernetes usando kubectl. Aprenderás lo principal sobre el CLI y como interactuar con tu aplicación

Un entorno Linux pre-configurado que puede ser usada como una consola (puedes escribir comandos). Haciendo click en los bloques de código el comando se ejecutará en la terminal.

Congratulations on completing this module

Now hear what you thought of the module

[Click here to give feedback](#)

[Sign in on Google](#) para guardar lo que llevas hecho.

[Sign in](#)

En el Módulo 3 > [_\(/docs/tutorials/kubernetes-basics/explore/explore-intro/\)](/docs/tutorials/kubernetes-basics/explore/explore-intro/)

2.3 - Consultar el estado de una aplicación

2.4 - Exponer una aplicación

2.5 - Escalar una aplicación

2.6 - Actualizar una aplicación

3 - Formación en línea

4 - Configuración

5 - Aplicaciones Stateless

6 - Aplicaciones Stateful

7 - Clústeres

8 - Servicios