# Create, Update & Delete Deployment

This example program demonstrates the fundamental operations for managing on [Deployment](#) resources, such as `Create`, `List`, `Update` and `Delete`.

You can adopt the source code from this example to write programs that manage other types of resources through the Kubernetes API.

## Running this example

Make sure you have a Kubernetes cluster and `kubectl` is configured:

```
kubectl get nodes
```

Compile this example on your workstation:

```
cd create-update-delete-deployment
go build -o ./app
```

Now, run this application on your workstation with your local kubeconfig file:

```
./app
# or specify a kubeconfig file with flag
./app -kubeconfig=$HOME/.kube/config
```

Running this command will execute the following operations on your cluster:

1. **Create Deployment:** This will create a 2 replica Deployment. Verify with `kubectl get pods`.
2. **Update Deployment:** This will update the Deployment resource created in previous step by setting the replica count to 1 and changing the container image to `nginx:1.13`. You are encouraged to inspect the retry loop that handles conflicts. Verify the new replica count and container image with `kubectl describe deployment demo`.
3. **List Deployments:** This will retrieve Deployments in the `default` namespace and print their names and replica counts.
4. **Delete Deployment:** This will delete the Deployment object and its dependent ReplicaSet resource. Verify with `kubectl get deployments`.

Each step is separated by an interactive prompt. You must hit the `Return` key to proceed to the next step. You can use these prompts as a break to take time to run `kubectl` and inspect the result of the operations executed.

You should see an output like the following:

```
Creating deployment...
Created deployment "demo-deployment".
-> Press Return key to continue.

Updating deployment...
Updated deployment...
-> Press Return key to continue.

Listing deployments in namespace "default":
 * demo-deployment (1 replicas)
```

```
-> Press Return key to continue.

Deleting deployment...
Deleted deployment.
```

## Cleanup

Successfully running this program will clean the created artifacts. If you terminate the program without completing, you can clean up the created deployment with:

```
kubectl delete deploy demo-deployment
```

## Troubleshooting

If you are getting the following error, make sure Kubernetes version of your cluster is v1.6 or above in `kubectl version`:

```
panic: the server could not find the requested resource
```