

# Chapter 11 Deploying a Stand-Alone Application

```
minikube start
```

```
minikube status
```

```
minikube dashboard
```

## Creating an App on dashboard

“+ Create”

- The application name is **webserver**
- The Docker image to use is **nginx:alpine**, where **alpine** is the image tag
- The replica count, or the number of Pods, is 3
- No Service, as we will be creating it later.

Click to go back, hold to see history

Search

+ CREATE

Resource creation

Cluster

- Namespaces
- Nodes
- Persistent Volumes
- Roles
- Storage Classes

Namespace

default

Overview

Workloads

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods

CREATE FROM TEXT INPUT

CREATE FROM FILE

CREATE AN APP

App name \*

webserver

9 / 24

Container image \*

nginx:alpine

Number of pods \*

3

Service \*

None

SHOW ADVANCED OPTIONS

DEPLOY

CANCEL

An 'app' label with this value will be added to the Deployment and Service that get deployed. [Learn more](#)

Enter the URL of a public image on any registry, or a private image hosted on Docker Hub or Google Container Registry. [Learn more](#)

A Deployment will be created to maintain the desired number of pods across your cluster. [Learn more](#)

Optionally, an internal or external Service can be defined to map an incoming Port to a target Port seen by the container. The internal DNS name for this Service will be: **webserver**. [Learn more](#)

Deploy a Containerized Application Web Interface

“Deploy”

```
$ kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
webserver	3/3	3	3	9m

```
$ kubectl get replicaset
```

NAME	DESIRED	CURRENT	READY	AGE
webserver-74d8bd488f	3	3	3	9m

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
webserver-74d8bd488f-dwbzz	1/1	Running	0	9m
webserver-74d8bd488f-npkzv	1/1	Running	0	9m
webserver-74d8bd488f-wvmpq	1/1	Running	0	9m

```
$ kubectl describe pod webserver-74d8bd488f-dwbzz
```

```
Name:          webserver-74d8bd488f-dwbzz
Namespace:     default
Priority:       0
Node:          minikube/10.0.2.15
Start Time:    Wed, 15 May 2019 13:17:33 -0500
Labels:        k8s-app=webserver
               pod-template-hash=74d8bd488f
Annotations:   <none>
Status:        Running
IP:            172.17.0.5
Controlled By: ReplicaSet/webserver-74d8bd488f
Containers:
  webserver:
    Container ID:  docker://96302d70903fe3b45d5ff3745a706d67d77411c5378f1f293a4bd721896d6420
    Image:         nginx:alpine
    Image ID:      docker-pullable://nginx@sha256:8d5341da24ccbdd195a82f2b57968ef5f95bc27b3c3691ace0c7d0acf5612edd
    Port:          <none>
    State:         Running
      Started:     Wed, 15 May 2019 13:17:33 -0500
    Ready:         True
    Restart Count: 0
  ...
```

kubectl describe pod <pod\_id>

## List Pods with Labels

```
$ kubectl get pods -L k8s-app,label2
```

NAME	READY	STATUS	RESTARTS	AGE	K8S-APP	LABEL2
webserver-74d8bd488f-dwbzz	1/1	Running	0	14m	webserver	
webserver-74d8bd488f-npkzv	1/1	Running	0	14m	webserver	
webserver-74d8bd488f-wvmpq	1/1	Running	0	14m	webserver	

```
$ kubectl get pods -l k8s-app=webserver
```

NAME	READY	STATUS	RESTARTS	AGE
webserver-74d8bd488f-dwbzz	1/1	Running	0	17m
webserver-74d8bd488f-npkzv	1/1	Running	0	17m
webserver-74d8bd488f-wvmpq	1/1	Running	0	17m

## Delete deployment

```
kubectl delete deployments webserver
```

- Deletes the ReplicaSet and Pods

## Deploy an App Using the CLI

Create `webserver.yaml` with:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webserver
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:alpine
          ports:
            - containerPort: 80
```

```
kubectl create -f web server.yaml
```

## Exposing an App

Create a `webserver-svc.yaml` with:

```
apiVersion: v1
kind: Service
metadata:
  name: web-service
```

```
labels:
  run: web-service
spec:
  type: NodePort
  ports:
  - port: 80
    protocol: TCP
  selector:
    app: nginx
```

Create the Service:

```
kubectl create -f web server-svc.yaml
```

Exposing a Deployment with the `kubectl expose` command:

```
kubectl expose deployment webserver --name=web-service --type=NodePort service/web-  
service exposed
```