

## Kubernetes

Step 1 :

Create a docker file for nginx and deploy it on docker hub or GCR.

Step 2 :

Create a kubernetes cluster and workload.

Step 3 :

Expose the microservice on ClusterIP, NodePort and LoadBalancer

The image shows two screenshots from a Google Chrome browser window. The top screenshot displays the Google Cloud Platform console for a Kubernetes Engine deployment. The left sidebar shows the navigation menu with 'Workloads' selected. The main content area shows the 'Deployment details' for a deployment named 'nginx-1'. It includes sections for 'Active revisions' (showing revision 10 as OK), 'Managed pods' (showing one pod running), and 'Exposing services' (showing three services: NodePort, ClusterIP, and LoadBalancer). The bottom screenshot shows a browser window with the URL '35.239.130.158' and the text 'nginx-1-6679bd5586-28wkq' displayed on the page.

Google Cloud Platform console screenshot showing Kubernetes Engine deployment details for 'nginx-1'.

Deployment details summary:

- Labels: app: nginx-1
- Logs: Container logs, Audit logs
- Replicas: 1 updated, 1 ready, 1 available, 0 unavailable
- Pod specification: Revision 10, containers: nginx

Active revisions table:

Revision	Name	Status	Summary	Created on	Pods running/Pods total
10	nginx-1-6679bd5586	OK	nginx: ketavbhatt/nginx2:v10	Jul 31, 2019, 4:27:07 PM	1/1

Managed pods table:

Revision	Name	Status	Restarts	Created on
10	nginx-1-6679bd5586-28wkq	Running	0	Jul 31, 2019, 4:27:07 PM

Exposing services table:

Name	Type	Endpoints
nginx-1-6mddl	NodePort	10.0.30.203:80 TCP
nginx-1-rgph4	ClusterIP	10.0.27.61
nginx-1-service	LoadBalancer	35.239.130.158:80

Autoscaler table:

Min/max replicas	1 / 5
Autoscaler	Autoscaler

Browser screenshot showing the service endpoint '35.239.130.158' and the text 'nginx-1-6679bd5586-28wkq'.